

**PATTERN THEORETIC
BAYESIAN INFERENCE
FOR MULTISENSOR FUSION**

FINAL REPORT
TO
ROME LABORATORY

CONTRACT NO. F30602-95-C-0171

DATED: FEBRUARY 1, 1996

Daniel H. Wagner, Associates, Incorporated
40 Lloyd Avenue, Suite 200
Malvern, Pennsylvania 19355
(610) 644-3400

Barry Belkin

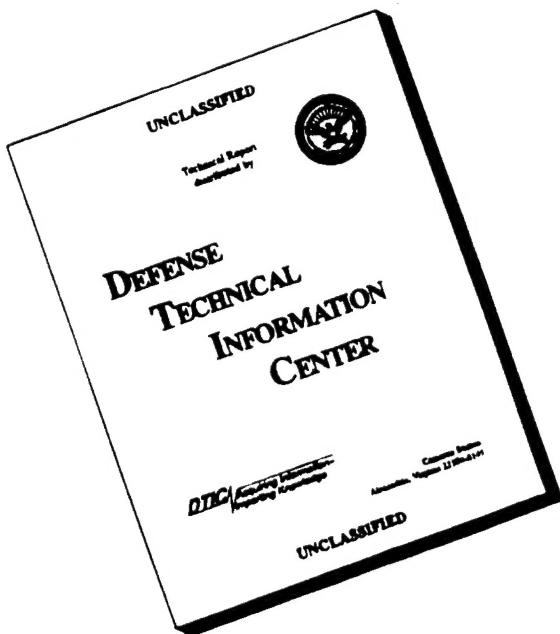
Barry Belkin

Stephan J. Suchower

Stephan J. Suchower

19960214 004

DISCLAIMER NOTICE



**THIS DOCUMENT IS BEST
QUALITY AVAILABLE. THE
COPY FURNISHED TO DTIC
CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO
NOT REPRODUCE LEGIBLY.**

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)			2. REPORT DATE February 1, 1996		3. REPORT TYPE AND DATES COVERED Final Report (30 May 95 - 31 Jan 96)		
4. TITLE AND SUBTITLE Pattern Theoretic Bayesian Inference for Multisensor Fusion			5. FUNDING NUMBERS C: F30602-95-C-0171				
6. AUTHOR(S) Barry Belkin and Stephan J. Suchower							
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Daniel H. Wagner, Associates, Incorporated 40 Lloyd Avenue, Suite 200 Malvern, PA 19355			8. PERFORMING ORGANIZATION REPORT NUMBER 6240				
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USAF/AFMC/Rome Laboratory Directorate of Contracting/PKRL 26 Electronic Parkway Griffiss AFB, NY 13441-4514			10. SPONSORING/MONITORING AGENCY REPORT NUMBER 0001AF				
11. SUPPLEMENTARY NOTES							
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; SBIR report , distribution unlimited			12b. DISTRIBUTION CODE				
13. ABSTRACT (Maximum 200 words) The focus of our Phase I study was to develop and test an algorithm to track a single aircraft of known type using the jump-diffusion method for state estimation. This technique avoids the limitations of conventional state estimation methods (most notably Kalman filtering) in dealing with the discrete state variables (such as target type) and the nonlinear/non-Gaussian measurements encountered in multisensor data fusion problems. The jump-diffusion technique operates by using Monte Carlo simulation to directly sample from the Bayes posterior distribution for the target state. The method is applied to the tracking of a maneuvering air target based on a combination of radar point tracking and optical imagery data. Numerical results show in particular that the aircraft orientation information extracted through the processing of optical images can significantly reduce tracking error relative to conventional point tracking methods. In Phase II we plan to continue the development of the Phase I algorithm by adding the capability to track multiple targets and to perform the automatic target recognition function when targets are of unknown type.							
14. SUBJECT TERMS multisensor data fusion, multitarget tracking, automatic target recognition, pattern recognition, image processing					15. NUMBER OF PAGES -49-		
					16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified		20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

INTRODUCTION	i
CHAPTER I: AUTOMATIC DETECTION, TRACKING AND RECOGNITION (ADTR)	
ALGORITHM DESCRIPTION.....	I-1
I-1 PHASE I ADTR PROBLEM DESCRIPTION	I-1
I-2 THE JUMP-DIFFUSION METHOD	I-2
I-2.1 THEORY	I-3
I-2.2 COMPUTATIONAL CONSIDERATIONS	I-8
I-3 ADTR ALGORITHM SPECIFICATION	I-11
I-3.1 AIRCRAFT DYNAMICS	I-11
I-3.2 STAGE 2 - HIGH RESOLUTION TRACKING	I-12
CHAPTER II: TEST STUDIES.....	
II-1 TEST CASE DESCRIPTIONS	II-1
II-1.1 TARGET AIRCRAFT TRACK	II-1
II-1.2 SENSOR CHARACTERISTICS	II-2
II-1.3 IMAGE TEMPLATES	II-4
II-2 NUMERICAL RESULTS	II-5
REFERENCES	R-1
APPENDIX	A-1

INTRODUCTION

This final report documents a Phase I SBIR project to implement and conduct initial testing of an algorithm to perform multisensor tracking of an aircraft based on the fusion of position measurements, angle measurements, and optical images of the aircraft. The work was performed by Wagner Associates under contract to Rome Laboratory, with consulting support from Dr. Michael Miller of the Department of Electrical Engineering at Washington University in St. Louis (WUSL).

The principal focus of the Phase I study was to adapt to a serial computing workstation environment (Tatung Super COMPstation 20) algorithmic procedures for executing automatic target detection, tracking, and recognition (ADTR) which were developed at WUSL and implemented across a network of Silicon Graphics workstations and a massively parallel DECMpp 12000/Sx (8192 processors).

We proceed to describe the general model framework of our analysis.

Sensors

The operational setting we assume in Phase I is that of a single aircraft of known type being tracked in combination by a low resolution radar which provides point estimates of the target location and a high resolution optical sensor which provides (i) measurements of the direction to the target and (ii) gray scale pixel images of the target body.

We model the low resolution sensor as providing relatively inaccurate measurements of target location ($\sigma_R = 200$ meters, $\sigma_{Az} = 0.5^\circ$, $\sigma_{El} = 2^\circ$). These measurement errors are intended to be representative of military air traffic control radars and shipboard surveillance radars.

The high resolution sensor provides more accurate target azimuth and elevation angle measurements ($\sigma_{Az} = \sigma_{El} = 0.1^\circ$) than the low resolution sensor but provides no direct range measurements. In addition, the high resolution sensor is assumed to generate focal plane pixel images of the target. A library of reference image templates has been provided to us by WUSL for an optical system. Each template in the library is a 64×64 pixel array with each pixel representing a signal intensity level discretized on a monochromatic (gray scale) of 256 possible levels.

Target Tracking

In the present context, a target track is the joint specification of the position, velocity, and orientation history of the target over a time window (possibly a moving window) and of the target aircraft type.

The ADTR algorithm automates the track estimation function by first applying classical Kalman methods to process the low resolution sensor data. The Kalman filter tracking solution is used to initialize the high resolution tracking.

The target image data provided by the high resolution sensor is processed in a quite different manner. In this case, a Monte Carlo scheme is used for random sampling from the Bayes posterior distribution on the joint position/velocity/orientation state of the target. The reported mean target state estimate and the associated error covariance are based on the sample statistics of this posterior distribution.

A principal objective of our Phase I study is to quantify the reduction in tracking error that derives from the fusion of the high resolution sensor data with the low resolution sensor data. There are two important effects at work here. First, the more accurate angle measurements provided by the high resolution sensor accelerate track solution convergence and reduce tracking error. Second, estimates of the target orientation angles (in particular, pitch and yaw) derived from the processing of image data can be used to draw inference about the instantaneous direction of target flight. Thus, there is the possibility of the early recognition of directional changes by the target. If only target position can be observed, the occurrence of a maneuver must be inferred through the measurement of subsequent target positions. The recognition of a target maneuver can then significantly lag its occurrence.

As noted, in Phase I we address only the single target tracking problem. In Phase II we plan to extend the ADTR algorithm to the multitarget tracking problem. We then expect to observe further benefits of multisensor fusion. Multiple targets are inherently easier to separate in problem space when

aircraft orientation and aircraft classification are included as state variables. Improved separation translates into a reduced incidence of report-to-track association errors and therefore into increased tracking accuracy.

Core Algorithm Design

In Chapter I we document the methods that underlie the ADTR algorithm. The basic approach is to use Monte Carlo simulation methods in conjunction with a random sampling procedure to construct sample target tracks which incorporate the fused location and image data provided by the sensors.

The sampling procedure we employ is based on the jump-diffusion model developed by Grenander and Miller (see reference [1]) and adapted in references [2] and [3] to the aircraft ADTR problem. The key idea is to allow a stochastic differential equation (SDE) and a Markov jump process to act in concert on the Monte Carlo samples of the target track. The role of the SDE is to act on the continuous target state variables (position, velocity, and orientation) and smoothly deform the Monte Carlo sample paths into sample tracks drawn from the Bayes posterior. The role of the jump process is to act on the single discrete target state variable (aircraft type) to allow new image data to alter the aircraft type assigned to an individual Monte Carlo track. (Note: Because we assume in Phase I that target type is known, we have deferred to Phase II the implementation of the jump feature of the jump-diffusion model.)

The SDE model (see equation (I-3)) effectively executes a randomized gradient descent search such that the term corresponding to the negative gradient of the Gibbs potential function drives the Monte Carlo paths in the direction of decreasing potential and hence increasing likelihood. The white noise term in the SDE adds the correct degree of randomness to the sampling process to generate the Bayes posterior distribution.

The underlying theory guarantees that the combined action of smooth target position, velocity, and orientation deformation by the SDE and the discrete jump process for assigning aircraft type will result in Monte Carlo tracks that closely approximate tracks drawn from the Bayes posterior. An essential computational advantage of this sampling procedure is that it accomplishes the sampling without requiring the computationally prohibitive task of constructing the posterior target track density function.

Numerical Studies

In Chapter II we summarize the numerical results we obtained based on our Phase I implementation of the ADTR algorithm. Our Phase I testing is focused exclusively on parametric studies to investigate the robustness of the tracking function of the algorithm as well as its sensitivity to certain key problem parameters related to the characteristics of the high resolution sensor and the characteristics of the templates in the image library. The following table identifies which problem parameters are varied and the assumed base case value of each.

<u>Parameter</u>	<u>Base Case Value</u>
High resolution sensor angular measurement error	0.1°
High resolution sensor data rate	2 Hz
Pixel resolution	64×64 pixels
Gray scale intensity discretization	256 levels
Pixel noise	12 gray scale units

The ground truth target track assumes an X29 aircraft initially in level flight at 10,000 ft. The aircraft then executes a banked (roll angle = -45°) and pitched (pitch angle = 20°) turn of 180° with a yaw rate of 5°/sec. The maneuver requires 36 seconds to complete.

We calculate and plot tracking error (measured as the distance between the estimated target position and the actual target position) versus time under three different assumptions about the available sensor data:

(LR): Low resolution position measurements only

(HRA): Low resolution position measurements + high resolution angle measurements

(HRI): Low resolution position measurements + high resolution angle measurements + high resolution pixel images.

These tracking accuracy comparisons are shown in Figures II-2 through II-11.

We summarize our base case numerical results as follows:

- **Case I:**
 - (i) target not maneuvering
 - (ii) fusion of low resolution (radar) and high resolution (optical) point tracking data (HRA processing)
 - (iii) Kalman filter state estimation methods.

Then tracking errors vary from 200 meters to 500 meters. This compares with tracking errors of as much as 3,000 meters under low resolution tracking (LR processing).

- **Case II:**
 - (i) target maneuvering (2g acceleration for 36 secs.)
 - (ii) fusion of low resolution point tracking data, high resolution point tracking data, and imaging data (HRI processing)
 - (iii) jump-diffusion method of state estimation.

Then tracking errors vary from 200 meters to 600 meters. This compares with tracking errors of as much as 1,400 meters under high resolution point tracking (HRA processing).

Phase I Study Conclusions

Our principal study conclusions are:

- We successfully demonstrated that the computations required to carry out the data fusion algorithm operations using our estimation methods, previously executed only within the context of a massively parallel computer architecture, can be performed on a serial processing workstation (the equivalent of a Sun Microsystems SPARC 20).

- Our study shows that there are significant benefits to be gained in terms of increased tracking accuracy through the fusion of multisensor point tracking and imaging data. We have demonstrated this specifically for the combination of a low resolution radar sensor and a high resolution optical imaging sensor. However, the methods we have developed to fuse multisensor data are quite general and can be extended to other sensing technologies.

Phase II Plan

In Phase II we will extend our analysis to the multiple target case. In that context the ADTR function will be extended to include data association, i.e., the formation, scoring, and pruning of hypotheses regarding which subsets of measurements to fuse into target tracks. The jump-diffusion method applied in Phase I to target type determination will be adapted to perform the critical data association operations.

Our planned Phase II technical objectives include:

- (1) Extend the application of the algorithm to the case of multiple air targets.
- (2) Expand the image template library to include multiple aircraft types and implement the program logic to perform the automatic target recognition function when the target type is unknown.
- (3) Allow for the aircraft images (both those stored in the template library and those observed by the optical sensor) to depend on visibility conditions and the level of ambient light.
- (4) Incorporate program logic to deal with unrecognizable targets. A given target may be unrecognizable for a number of reasons:
 - The target is an aircraft of a type not included in the image template library
 - Multiple targets are closely spaced causing some degree of image occlusion
 - Camouflage is being employed in an attempt to defeat a tracking system based on image recognition methods.
- (5) Extend the application of the algorithm to microwave and millimeter wave radar imaging technologies including SAR and ISAR.

CHAPTER I

Automatic Detection, Tracking and Recognition (ADTR) Algorithm Description

In this chapter we document the algorithm to perform the automatic detection, tracking, and recognition (ADTR) function which we have implemented as part of our Phase I study. Mathematical details are included to the extent that they do not already appear in published papers and reports (see, for example, references [3], [4], and [5]) and to the extent that such details contribute to an understanding of how the underlying theory has been applied in the specific context of our Phase I study.

As noted in the Introduction, the focus of our Phase I study has been the implementation and initial testing of an algorithm to execute the function of the automatic recognition of an air target. We begin in Section I-1 with a formal statement of the ADTR problem addressed in Phase I. In Section I-2 we describe the random sampling technique based on the jump-diffusion process model which underlies our ADTR algorithm. The specific ADTR algorithm procedures are detailed in Section I-3.

I-1 Phase I ADTR Problem Description

In defining the automatic target recognition problem, we distinguish between those components of the target state vector which vary continuously and those components which vary discretely. In the context of our Phase I study, the continuous state components are taken to be the aircraft position vector $p = (p_1, p_2, p_3)$, the aircraft velocity vector $v = (v_1, v_2, v_3)$, and the aircraft orientation (Euler) angle vector $\phi = (\phi_1, \phi_2, \phi_3)$, as specified in an inertial coordinate frame. Here, ϕ_1 = roll angle, ϕ_2 = pitch angle, and ϕ_3 = yaw angle. The single discrete state variable is aircraft type, denoted by $a \in A$, where A is a known alphabet of possible aircraft types.

We postulate two types of sensors that provide information about the target state vector. First, there is a low resolution tracking sensor that provides measurements of target position. We model the low resolution sensor as generic in that the particular sensing technology it employs is unimportant. The only essential requirement is that each sensor measurement provide an estimate of target azimuth angle, elevation angle, and range. Measurement errors in the three coordinates are assumed to be statistically independent. Possibilities for the low resolution sensor are (i) a narrow band passive cross-array of isotropic sensor elements (see reference [3]) and (ii) a tracking radar.

Second, there is a high resolution imaging sensor that provides pixel images of the target such that the signal in each pixel is a noisy gray scale intensity level. The case we treat in Phase I is that of an optical sensor system that produces 2-D far-field orthographic projections of the target image on the sensor focal plane. Each projected image is represented by a 64×64 array of gray scale intensity levels (discretized as integer values on the interval $[0, 255]$). Thus, a high resolution measurement is an element of the space $[0, 255]^{64 \times 64}$.

In our Phase I problem formulation, we assume that the low and high resolution sensors are collocated at a single ground tracking station. The extension to sensors at multiple geographic locations is deferred to Phase II.

The ADTR problem, as we have posed it in Phase I, then is to develop an algorithmic procedure to detect a single air target, identify its type (i.e., classify the target), and generate estimates for its position, velocity, and orientation. This combination of detection, tracking, and recognition functions is to be performed on the basis of a combination of low resolution and high resolution sensor data, together with prior knowledge about the candidate aircraft types and flight characteristics.

I-2 The Jump-Diffusion Method

The approach we take to the ADTR problem is based on the jump-diffusion method for sampling from the Bayes' posterior density for the target track given a combination of prior information about the target and accumulated sensor data. In the current context, a target track $\mathbf{X}(t)$ is a specification of the target path through position, velocity, and orientation space at a sequence of discrete times t_k over the time interval $[0, t]$ as well as a specification of the target type. We take the t_k to be the sensor measurement times.

I-2.1 Theory. As we describe below, a jump-diffusion process has the defining properties that it (i) executes jumps in the discrete state variables at exponentially distributed times and (ii) follows a stochastic differential equation diffusion model in the continuous state variables between the jumps. The key theoretical result on which our state estimation methods are based is an ergodic theorem proven in reference [2] which insures that the jump-diffusion process we construct converges in distribution to the Bayes' posterior distribution for the target track conditioned on all past information (both prior information and sensor measurements). This result provides the basis for a computational procedure for drawing Monte Carlo sample target paths. The mean target track and the estimation error covariance are approximated as sample statistics of the set of Monte Carlo tracks.

As we have posed the ADTR problem, there is a time (denoted by t_1) at which the initial high resolution measurement is made and the tracking problem transitions from one of six-state position/velocity estimation to one of full-state (10-state) estimation. We let t_0 denote the time of the last low resolution measurement prior to t_1 . We refer to the low resolution tracking phase ending at t_0 as *stage 1* and to the joint low resolution and high resolution tracking phase beginning at t_1 as *stage 2*.

Standard Kalman filtering methods are adequate for stage 1 tracking. The assumption then is that at time t_0 a six-state filter estimate of the target position and velocity (and hence the direction of target flight) is available together with an associated 6×6 error covariance matrix. As described in Section I.3, the stage 1 estimate of aircraft direction is used to draw statistical inference about the target aircraft orientation angles. The result is that the stage 1 filter output provides the basis to construct a joint distribution on target position, velocity, and orientation to initiate the *stage 2* (high resolution) tracking phase.

The state estimation technique we use in stage 2 is fundamentally different from that used in stage 1. In stage 2 we use a form of Gibbs sampling that enables random draws to be made from the posterior target track likelihood function without the need to construct the full likelihood surface.

The particular form of Gibbs sampling that we use (see reference [3]) is based on the Gibbs representation of the Bayes' posterior measure $\mu_t(dx)$:

$$\mu_t(dx) = \frac{e^{-H_t(x)}dx}{\int e^{-H_t(y)}dy}. \quad (\text{I-1})$$

The function H_t is referred to as the potential function associated with $\mu_t(dx)$. We observe that up to an additive constant H_t is simply the negative log-likelihood associated with the Bayes' posterior measure.

In the current application, H_t is defined on the space of possible target tracks $X(t)$ on $[0, t]$ sampled at a sequence of discrete times. In general, it can then be shown that H_t has an additive decomposition of the form $H_t = \sum \phi_i$, such that each element of prior information about $X(t)$ and each element of information about $X(t)$ provided by a particular sensor measurement separately give rise to a summand ϕ_i in the potential function decomposition.

The component likelihoods which contribute to $H_t(X(t))$ are

- the prior joint likelihood of the sequence of target position/velocity vectors associated with the track $X(t)$
- the prior joint likelihood of the sequence of target Euler angle vectors associated with the track $X(t)$
- the likelihood of measuring the target position at time t_k to be z_k given target track $X(t)$
- the likelihood of measuring the target pixel image at time t_k to be I_k given target track $X(t)$.

In Section I-3 we will derive expressions for the above likelihood function components as well as expressions for the associated potential function components ϕ_i . In the case of position and velocity, we will postulate a stochastic differential equation from which the required prior likelihood can be derived. We will also postulate a Markov model for the time evolution of target orientation. In this case, the target orientation model will define one likelihood function component. A second component will be defined by the constraint that the target pitch and yaw angles must be consistent (in a statistical sense we will make precise) with the direction of the target velocity vector.

For the particular sensors modeled in our Phase I study, the ϕ_i arising from sensor measurements can all be dealt with within the common framework of measurements with Gaussian error. In that case, the corresponding potential function contributions take the general form

$$\phi_i(x) = (\hat{z} - z_p(x))^T \Sigma^{-1} (\hat{z} - z_p(x)), \quad I-2)$$

where

\hat{z} = observed measurement vector

$z_p(x)$ = predicted measurement vector conditioned on target state x

Σ^{-1} = error covariance of observed measurement vector.

The quantity on the RHS of equation (I-2) is the so-called Mahalonobis distance, measuring in a statistical sense the distance between the predicted measurement $z_p(x)$ conditioned on the assumed target state x at the time of the measurement and the observed measurement \hat{z} .

In the case of a low resolution sensor measurement, $z_p(x)$ is the target position. Hence, there is a simple analytic relationship (projection onto a subspace) between the target state x and the predicted measurement. The corresponding potential function term $\varphi_i(x)$ is therefore also a known function of x , and the required differentiations in calculating the gradient of the potential function can be carried out analytically.

In the case of an image produced by the high resolution optical sensor, the function $z_p(x)$ depends in a complex way on the target state. The values of $z_p(x)$ are known only numerically and are stored in a precomputed template library. Consistent with our model for the optical images produced by the high resolution sensor, each template image is represented by a 64×64 array of gray scale intensity levels (discretized as integer values on the interval [0,255]). A separate template is stored for each combination of aircraft orientation and aircraft type. For this purpose, roll angle and pitch angle are each discretized in 10 degree increments. The step size in yaw angle varies with pitch angle in such a way that the density of grid points on the yaw-pitch sphere is approximately uniform.

All of the stored image templates are based on a reference image size scale and a reference azimuth-elevation coordinate origin in the focal plane of the imaging sensor. The tracking filter generates the target range estimate to rescale each observed optical image to the reference range. We assume that background pixels can be distinguished from aircraft body frame image pixels and take the center of area of the projected target image as the coordinate origin. In the context of our numerical studies, we simulate aircraft pixel images directly in the template library viewing frame and thereby bypass the image registration step.

We model the noise in each pixel as additive Gaussian and assume statistical independence from pixel to pixel. The combination of the template library, the Gaussian noise model, and simulated image data provides the

basis for determining the ρ_i associated with each high resolution measurement.

We note that the template library used in our Phase I study was generated by Dr. Michael Miller and his colleagues of the Department of Electrical Engineering at Washington University in St. Louis (WUSL). The images in the library are all for a standard set of optical viewing conditions with regard to atmospheric clarity and background light intensity level. In Phase II we plan to expand the template library to include a range of viewing environments.

The jump-diffusion method proceeds by writing down a stochastic differential equation (SDE) of the Langevin type involving the gradient ∇H_t of the posterior potential function:

$$dX(t) = -\frac{1}{2}QQ^T \nabla H_t(X(t))dt + QdW(t), \quad (I-3)$$

where $dW(t)$ is a standardized white noise process of the appropriate dimension and the matrix Q governs the scale of the diffusion. In application Q is taken to be a diagonal matrix. This SDE defines a continuous path stochastic process describing a randomized gradient descent on the posterior potential function $H_t(X(t))$.

The jump element of the model comes into play in the recognition of the target type. Target type is a discrete variable, and a mechanism different from the continuous track deformation implied by equation (I-3) is required to allow transitions in target type. Specifically, it must be possible for the target type assigned to a Monte Carlo sample track to change abruptly from a to b such that the probability of such a change is determined by the relative weight of statistical evidence favoring b as the target type as opposed to a .

The particular scheme we use for determining when such discrete changes in target type occur is based on an adaptation of the Metropolis algorithm (see reference [6]). A key relevant quantity is

$$H_t(X(t_k)|a) = \text{the posterior potential of } X(t_k) \text{ given target is of type } a.$$

The decision about whether to alter the target type assigned to a particular sample track is based on $H_t(X(t_k)|a)$.

Suppose that the target type assigned to the sample track $X(t_k)$ at time t_k is a_0 and that the alternative target type possibilities are a_1, a_2, \dots, a_I . A random draw is made from the set of alternative target types based on their relative likelihood of occurrence. Suppose that target type a_i is drawn. Then the rule for assigning the new target type b to $X(t_k)$ is

$$b = a_i \quad \text{if } H(X(t_k)|a_i) \leq H(X(t_k)|a_0). \quad (\text{I-4})$$

Otherwise,

$$b = \begin{cases} a_i & \text{with probability } e^{-[H_{t_k}(X(t_k)|a_0) - H_{t_k}(X(t_k)|a_i)]} \\ a_0 & \text{with probability } 1 - e^{-[H_{t_k}(X(t_k)|a_0) - H_{t_k}(X(t_k)|a_i)]}. \end{cases} \quad (\text{I-5})$$

As previously noted, it is shown in reference [2] that the jump-diffusion model has certain ergodic properties which insure that the process has a limiting stationary distribution equal to the posterior distribution on the target track conditioned on all past information (both prior information and sensor data). This result provides the theoretical basis for the discrete sampling procedure we now describe for estimating the target track.

Let $t_k, k=1,2,\dots,K$, denote the stage 2 measurement times. Then consider the stochastic difference equation which approximates the stochastic differential equation in equation (I-3):

$$\Delta X(t) = -\frac{1}{2} Q Q^T \nabla H_t(X(t)) \Delta t + Q \Delta W(t). \quad (\text{I-6})$$

The required sample paths are initialized at t_0 by making N independent draws from the joint distribution on target position and orientation based on the time t_0 tracking solution derived from the output of the stage 1 tracking filter. The assignment of a target type to each path is also made on the basis of Monte Carlo draws, consistent with the (assumed known) frequency function for occurrence of target type. We note that our numerical results in Chapter II are based on $N = 100$ sample paths.

Once the N target tracks have been initialized, the next operation is to extend each track from t_0 to t_1 . Two steps are involved: (i) the forward projection of each track from time t_0 to time t_1 and (ii) the updating of each

track for the time t_1 measurement data. The forward projection step for each path is based on a dynamical model for predicting the target track at t_1 conditioned on the sensor data through t_0 . This dynamical model is described in detail in Section I-3.1.

The measurement update at time t_1 is accomplished through the application of the stochastic difference equation in equation (I-6). An essential element of the theory in equation (I-6) is that it must be iterated multiple times to incorporate the effect of the measurement at t_1 into the target track estimate. Each iteration is referred to as a *cycle*. A sufficient number of cycles M must be applied so that the target track arrived at under the action of the diffusion equation represents a random draw from the Bayes posterior distribution on the target state conditioned on all of the past data.

The same two step process is applied in updating the sample tracks from time t_i to time t_{i+1} . First, the dead-reckoned estimate of the target track at t_{i+1} is determined based on the dynamical model for the target state. Then the jump-diffusion model is applied at t_{i+1} to incorporate the measurement at t_{i+1} into the state estimate.

I-2.2 Computational Considerations. The theory provides no explicit guidance on how large M must be at each t_i or how small the discrete steps Δt at each cycle must be. Our experience with the numerical application of the diffusion model, however, has provided the following insights:

- In applying the diffusion equation, the matrix QQ^T in equation (I-6) effectively scales the step size in each coordinate and provides the option to choose a different step size for each of the three categories of state variable: position, velocity, and orientation. In each case, Δt is chosen so that the change in state induced by a single diffusion cycle is small compared to the precision with which the mean target track is to be determined. The numerical results in Chapter II are based on the values:

$$\begin{aligned}\Delta p_{\max} &= 5 \text{ meters} \\ \Delta v_{\max} &= 1 \text{ meter / sec} \\ \Delta \phi_{\max} &= 2.5^\circ.\end{aligned}\tag{I-7}$$

In each case, the corresponding Δt is obtained by substituting the value of $-\frac{1}{2} \nabla H_t(\mathbf{X}(t))$ for the current cycle into equation (I-6), assuming the contribution of the white noise term to be $\sqrt{\Delta t}$ and solving the resulting quadratic in Δt . The effect then is that

Δt is dynamically adjusted for each category of state variable following each cycle.

- There is a tendency for a very large number of cycles to be required to achieve solution convergence in equation (I-6) if the posterior mean track estimate at t_i differs significantly from the prior mean track at t_i . This situation occurs when a measurement with significant corrective power (high information content) such as the initial high resolution measurement is processed. In that case, it becomes necessary to significantly increase the tolerances in equation (I-7) to accelerate convergence. The standard tolerances are then restored for subsequent update times.

In obtaining the numerical results in Chapter II, we used $M = 2,000$ cycles except in processing the initial high resolution measurement at t_1 . In that single instance, we set $M = 5,000$.

We observe that the principle at work in our random sampling technique based on equation (I-3) involves the interaction between the gradient of the potential term and the white noise term. The $-\frac{1}{2}\nabla H_t$ term tends to deform the sample path so as to reduce its Gibbs potential and hence to increase its likelihood. The white noise term injects randomness. This accomplishes two things. First, it prevents the path from getting stuck at a local minimum of the potential function. Second, it injects the correct amount of randomness into the diffusion process to assure that the cycling process generates sample paths approximately in proportion to their relative likelihood.

One very important advantage of the jump-diffusion method is that the sampling of the likelihood function is accomplished without having to build the entire likelihood surface. While the jump-diffusion method can require significant computation, the amount of computation required to construct the full likelihood surface, because of the high dimensionality of the target track state space, would be prohibitive.

The application of equation (I-3) requires the calculation of the gradient of the potential function. Because of the additive form of H_t , the quantity $-\frac{1}{2}\nabla H_t$ can be expressed in terms of the quantities $\nabla \phi_i$. Analytic expressions can be derived for all of the $\nabla \phi_i$ except for the terms arising from the target pixel images. In that case, the ϕ_i are specified numerically based on the pixel images in the template library. A first-order finite difference method is then used to calculate each $\nabla \phi_i$.

In Section I-3 below, we will give explicit expressions for each of the potential terms ϕ_i . Let $x(j) = (p(j), v(j))$ denote the target position/velocity at time t_j . Then one term of the potential function at time t_k denoted $\phi_i(x(1), x(2), \dots, x(K))$ arises from the joint distribution of the target position and velocity at the observation times t_1, t_2, \dots, t_K conditioned on the past data through time t_{K-1} . The calculation of $\nabla_{x(K)}\phi_i$ requires knowledge of the covariance matrix for $x(K)$ conditioned on the data through t_{K-1} . The sample covariance for $x(K)$ obtained from the M Monte Carlo sample paths at t_K but prior to processing the observation at t_K is used as an estimate for this covariance. The value $M=100$ was chosen with this requirement in mind.

We observe that as the time index k increases, the amount of computation required to execute a tracking update grows at an approximately linear rate. One way of inhibiting this growth is to restrict the segment of the target track that is updated to a moving window. If the length of this window is set to include the L most recent high resolution measurements, then the measurement at time t_k would be used to modify the earlier target states only at those earlier measurement times t_j such that $k-j \leq L-1$. The result of employing such a moving window is that the amount of computation grows linearly only up until the time of the L^{th} high resolution measurement. Beyond that point, the computational load per update remains roughly constant.

The results in Chapter II are based on a window of length one. Our estimation methods then constitute a form of filtering — the estimation of the current system state conditioned on all past data. We note that the algorithm code has been written to handle processing windows of any user specified length. In its more general mode of operation, the algorithm can be used to perform the smoothing function over a moving window of fixed length or over the entire track history.

We observe that the jump-diffusion approach to sampling from the posterior distribution on the target track differs fundamentally from conventional approaches to Monte Carlo tracking. The usual procedure begins by postulating a prior distribution on the space of possible tracks. Monte Carlo samples are drawn from this prior. These sample tracks are then assigned relative weights based on the likelihood of the observed sensor measurements given each track. As more and more measurements are used to refine the sample tracks, the most unlikely tracks are pruned while those that survive are replicated (the regeneration process) to insure adequate statistical coverage in the shrinking neighborhood in track space to which ground truth becomes progressively confined.

Thus, a basic feature of a conventional Monte Carlo tracker is that the refinement of track estimates is achieved through the operations of pruning and regeneration. Individual sample tracks are extended forward in time but otherwise do not change in response to new data. What does change are the relative weights assigned to the sample tracks.

By contrast, in the jump-diffusion model approach all sample tracks receive the same relative weight. It is the tracks themselves which are deformed in response to the data through a combination of continuous variation and discrete jumps. It is this deformation process which governs the statistical properties and convergence properties of the sample track estimates.

The tracking and identification of an aircraft leads to a complex posterior likelihood function and a high-dimensional state estimation problem. In such an application, the inherent efficiency of the path deformation method in concentrating the Monte Carlo sample paths in the relevant region(s) of track space is a critical factor in choosing that method over the path re-weighting method.

We now turn in Section I-3 to a more detailed specification of our ADTR algorithm than we have attempted to this point.

I-3 ADTR Algorithm Specification

In separate subsections below we describe the model used for aircraft dynamics and the stage 2 (high resolution) ADTR algorithm.

I-3.1 Aircraft Dynamics. We begin with a brief description of our model for aircraft dynamics. Additional details are provided in the Appendix.

As previously described, the instantaneous dynamical state of an aircraft is represented in inertial coordinates in terms of a nine-vector comprised of a three-vector $p = (p_1, p_2, p_3)$ specifying the aircraft position, a three-vector $v = (v_1, v_2, v_3)$ specifying the aircraft velocity, and a three-vector $\phi = (\phi_1, \phi_2, \phi_3)$ of Euler angles specifying the aircraft orientation (ϕ_1 = roll angle, ϕ_2 = pitch angle, ϕ_3 = yaw angle). Roll angle and yaw angle each take values on the interval $[0, 2\pi]$ with the endpoints 0 and 2π identified. Pitch angle takes values on the interval $[-\pi/2, \pi/2]$. The aircraft track state $X(t)$ at time t specifies the continuous time state history over the time interval $[0, t]$ and therefore

$$X(t) \in (\mathbb{R}^6 \times [0, 2\pi]^2 \times [-\pi/2, \pi/2])^{[0, t]}.$$

The particular inertial frame we choose is the topocentric coordinate frame centered at the common location of the low resolution sensor and the high resolution sensor. The positive x-direction is local east, the positive y-direction is local north, and the positive z-direction points in the direction of the outward normal to the Earth surface. Azimuth angle, elevation angle, and range are the spherical coordinates in this topocentric frame. In the context of our tracking problem, the position and velocity of the aircraft are measured relative to the rotating Earth. This allows us to treat the Earth as stationary and the topocentric frame as inertial.

The aircraft orientation angles are defined by the sequence of axis rotations defining the transformation from the aircraft body frame, which has its coordinate axes fixed along the principal axes of the airframe, back into the inertial frame. This transformation is given by

$$\Psi_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_1) & \sin(\phi_1) \\ 0 & -\sin(\phi_1) & \cos(\phi_1) \end{bmatrix} \times \begin{bmatrix} \cos(\phi_2) & 0 & -\sin(\phi_2) \\ 0 & 1 & 0 \\ \sin(\phi_2) & 0 & \cos(\phi_2) \end{bmatrix} \times \begin{bmatrix} \cos(\phi_3) & \sin(\phi_3) & 0 \\ -\sin(\phi_3) & \cos(\phi_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (I-8)$$

The transformation Ψ_{ϕ} plays an important role in the use of the image library. The templates as catalogued are for a standardized viewing geometry which places the observer at $(0,1,0)$ (i.e., at unit distance along the positive y-axis). In general, if an aircraft with orientation ϕ is viewed at azimuth θ and elevation φ , then the appropriate orientation ϕ' for purposes of library look-up satisfies the relation $\Psi_{\phi'} = \Psi_{\phi} R(\theta, \varphi)$, where $R(\theta, \varphi)$ rotates the inertial frame into the viewing frame.

I-3.2 Stage 2 – High Resolution Tracking. We consider the stage 2 case in which there is a current time t_0 through which the processing has proceeded. We let t_1, t_2, \dots, t_j denote a sequence of subsequent times and describe the procedure for extending the processing to t_j . We note that not all of the t_j need be measurement times.

We describe the stage 2 procedure for the general case where the joint distribution on target position and velocity at time t_0 is multivariate Gaussian. At the start of stage 2, t_0 is taken to be the time of the last low resolution sensor measurement prior to stage 2. High resolution tracking is then initialized based on the multivariate Gaussian track solution output at

t_0 by the stage 1 tracking filter. Throughout the subsequent stage 2 processing, Monte Carlo methods are used. The role of t_0 is then played by the start time of the time window over which the sample target tracks are to be updated for sensor data.

We proceed to describe the calculation of the various terms ρ_i contributing to the potential function. We consider first the term arising from prior information on the aircraft Euler angles.

The approximate values $\bar{\phi}_2$ and $\bar{\phi}_3$ of the initial pitch angle and initial yaw angle of the aircraft can be inferred from the direction of $v(t_1)$. Assume that the actual initial pitch angle is constrained to the interval $[\bar{\phi}_2 - \epsilon_2, \bar{\phi}_2 + \epsilon_2]$ and the actual initial yaw angle is constrained to the interval $[\bar{\phi}_3 - \epsilon_3, \bar{\phi}_3 + \epsilon_3]$. No information is assumed known about the initial aircraft roll angle other than that it is constrained to the interval $[0, 2\pi]$.

Under these conditions, the probability density on the initial triple of Euler angles (ϕ_1, ϕ_2, ϕ_3) is taken to have the form (for $\pi/2 - |\bar{\phi}_2| >> \epsilon_2, \epsilon_3$):

$$g(\phi_1, \phi_2, \phi_3) = \begin{cases} \frac{\cos(\phi_2)}{8\pi\epsilon_3 \cos(\bar{\phi}_2) \sin(\epsilon_2)} & \text{for } \phi_1 \in [0, 2\pi], \phi_i \in [\bar{\phi}_i - \epsilon_i, \bar{\phi}_i + \epsilon_i], i = 2, 3 \\ 0 & \text{otherwise.} \end{cases} \quad (\text{I-9})$$

We use the Von-Mises maximum entropy distribution on the circle to model prior uncertainty. The prior conditional density on $\phi(j)$ given $\phi(j-1)$ for $j \geq 2$ is thus assumed to have the Markov Von-Mises form

$$\prod_{i=1}^3 \frac{1}{2\pi I_0(\kappa_i)} e^{\kappa_i \cos[\phi_i(j) - \phi_i(j-1)]}. \quad (\text{I-10})$$

Here, $\kappa = (\kappa_1, \kappa_2, \kappa_3)$ is a vector of parameters characterizing the degree of concentration of Euler angle prior density. We assume that each κ_i is of the form $\kappa_0(\Delta t)^{-1}$, where Δt is the length of the prediction time step. The scale parameters κ_0 for roll, pitch, and yaw are model inputs.

It follows from equation (I-10) that the contribution to the potential function arising from the Markov transition model for the aircraft Euler angles has the form

$$\phi_i(\phi(1), \phi(2), \dots, \phi(K)) = - \sum_{j=1}^K \sum_{l=1}^3 \kappa_l \cos(\phi_l(j) - \phi_l(j-1)). \quad (I-11)$$

We next turn to the potential term arising from the position/velocity model.

We assume a stochastic differential equation model for aircraft position and velocity of the particularly simple form

$$\begin{cases} dp = v dt \\ dv = B dW(t), \end{cases} \quad (I-12)$$

where $dW(t)$ is a three-dimensional white noise process scaled by B to produce random accelerations with covariance matrix BB^T . We assume that the acceleration process is isotropic, so that BB^T has the special form $\sigma^2 I$. (In Phase II we will consider the case of non-isotropic random accelerations.)

The SDE in equation (I-12) is linear and lends itself to the standard recursive estimation methods of Kalman filtering. Let $x(j) = (p(j), v(j))$ and assume that $x(0)$ is given. Under the action of the dynamical model in equation (I-12), the conditional distribution for $x(j+1)$ given $x(j)$ is multivariate Gaussian. For $j = 0, 1, \dots, K-1$ define

$$\begin{aligned} \mu_{j+1} &= E[x(j+1)|x(j)] \\ \Sigma_{j+1} &= \text{Cov}[x(j+1), x(j+1)|x(j)]. \end{aligned} \quad (I-13)$$

Then $\mu_{j+1} = A_j x(j)$, where the matrix A_j can be expressed in terms of the coefficients in equation (I-12). The Kalman filter prediction equations also provide a recursive relationship for covariance matrices Σ_j .

It follows that the potential function contribution ϕ_i associated with the joint distribution of the $x(j)$, $j = 1, 2, \dots, K$, can be decomposed as

$$\varphi_i(x(1), x(2), \dots, x(K)) = \sum_{j=1}^K (x(j) - \mu_j)^T \Sigma_j^{-1} (x(j) - \mu_j). \quad (I-14)$$

It is an easy matter to calculate $\nabla \varphi_i$ from the above expression.

The next potential term arises from the assumed relationship between the direction of the aircraft velocity vector and the aircraft orientation angles. Let

$$\begin{aligned} \theta(j) &= \text{azimuth angle of } v(j) \\ &= \tan^{-1} \left[\frac{v_2(j)}{v_1(j)} \right] \\ \varphi(j) &= \text{elevation angle of } v(j) \\ &= \tan^{-1} \left[\frac{v_3(j)}{\sqrt{v_1^2(j) + v_2^2(j)}} \right]. \end{aligned} \quad (I-15)$$

(Note: We assume that the aircraft velocity vector provides no information about the aircraft roll angle.)

We then let

$$\varphi_i = - \sum_{j=1}^K [\kappa_{el} \cos(\varphi(j) - \phi_2(j)) + \kappa_{az} \cos(\theta(j) - \phi_3(j))], \quad (I-16)$$

where κ_{el} and κ_{az} are constants chosen to fit the Von-Mises density to the density $g(\phi_1, \phi_2, \phi_3)$ in equation (I-8). We note that the effect of this φ_i term on the overall potential function is mathematically equivalent to that of a measurement of the direction of the aircraft velocity vector given the aircraft pitch and yaw or, equivalently, a measurement of the aircraft pitch and yaw given the direction of the aircraft velocity vector.

Monte Carlo sample tracks are extended from t_k to t_{k+1} in two stages. The first stage is to randomly draw the orientation angles at t_{k+1} . This is done using the Von Mises density in equation (I-10) governing orientation angle transitions. The second stage is to randomly draw the aircraft position and

velocity at t_{k+1} based on equations (I-12) and (I-13). Once the Monte Carlo tracks have been projected forward to t_{k+1} , the diffusion model is applied to incorporate the effect of sensor data.

A summary of the steps involved in the execution of stage 2 of the ADTR algorithm are as follows:

- Step 1.** Initialize N Monte Carlo tracks at time t_0 by making random draws, first from the aircraft position/velocity density (based on the low resolution tracking filter) and then from the conditional density on the aircraft orientation (equation (I-8)). Set $k = 0$.
- Step 2.** Project the time t_k Monte Carlo tracks forward in time to t_{k+1} based on the SDE model for position/velocity and the Von-Mises model for orientation. Set $k = k + 1$.
- Step 3.** Process the sensor measurements at times t_1, t_2, \dots, t_k using equation (I-3) and a random sampling method based on the jump-diffusion model. Execute M sampling cycles for each Monte Carlo track $X(t_k)$, reevaluating $\nabla H_t(X(t_k))$ after each cycle. Return to step 2.

The above procedure proceeds iteratively until all of the scheduled updates are executed or until there are no more measurements to be processed. After the measurement update is completed at each t_k , the following sample statistics are calculated and output:

- (i) the sample means for the aircraft position vector, velocity vector, and orientation vector
- (ii) the three submatrices of the error covariance matrix corresponding to position, velocity, and orientation
- (iii) the fraction of Monte Carlo paths assigned each of the possible target types.

In our Phase I implementation of the algorithm, we assume the target aircraft to be of known type and focus on the tracking function. The numerical results we present in Chapter II compare the sample mean estimate for target position and the (simulated) ground truth target position. The magnitude of the vector difference between the estimated and ground

truth target positions is the assumed measure of tracking error. Our interest then is in how tracking error behaves as a function of time, particularly during a target maneuver.

We note that the algorithm we have implemented in Phase I differs from that described in reference [3] in certain important respects.

The algorithm described in reference [3] tracks the target in the six-state space of position and orientation. We have added velocity to the target state vector. Since the translational motion of the aircraft is a Markov process jointly in position and velocity, the state prediction step can then be performed recursively in much the same manner as in Kalman filtering. Furthermore, when velocity is explicitly part of the target state, the relationship between the target (translational) direction of motion and the target orientation can be enforced directly through the Gibbs potential function.

The algorithm in [3] merges initial detection and the initiation of high resolution tracking into a single integrated operation. The point of view we have taken is that because the high resolution sensor has a narrow field of view, in an operational setting it is likely that track initiation would be performed by a low resolution search sensor with a much wider field of view. High resolution tracking would then be initiated via a direct handoff from the low resolution sensor. We have taken this two-stage approach in modeling the tracking process. A computational benefit is that it eliminates the computation intensive function of having to search through the entire template library to initiate high resolution tracking.

CHAPTER II

Test Studies

In this chapter we summarize the results of our numerical studies applying the tracking algorithm. In Section II-1 we define the test cases we analyzed. In Section II-2 we present numerical results and in Section II-3 discuss our Phase I study conclusions.

II-1 Test Case Descriptions

The test cases we ran were designed with two objectives in mind. First, we wanted to quantify the benefits of multisensor tracking. In particular, we wanted to demonstrate that the error in tracking a target aircraft can be reduced significantly by fusing high resolution optical image data with low resolution radar data. Second, we wanted to investigate the robustness of the tracking accuracy of the algorithm to variations in the characteristics of the high resolution sensor as well as to variations in the resolution and accuracy of the image template library.

We first describe the assumed ground truth aircraft track. We then identify the assumptions made about the sensors and the image library.

II-1.1 Target Aircraft Track. In general, we define a ground truth aircraft track in terms of the following parameters:

- initial aircraft position vector $p(0)$ and orientation angle vector $\phi(0)$
- aircraft speed \bar{v} (assumed to be constant)

- a sequence of time intervals $[t_i, t_{i+1}]$, $i = 0, 1, \dots, I-1$, (with $t_0 = 0$) and orientation angle rate vectors $\dot{\phi}(i)$ such that $\dot{\phi}(i)$ is the assumed constant aircraft orientation angle rate vector on $[t_i, t_{i+1}]$.

We assume that at any instant the direction of aircraft flight is the positive x-direction in the body coordinate frame. As a consequence of this assumption, the above data are sufficient to determine the full nine-dimensional position, velocity, and orientation state of the aircraft as a function of time.

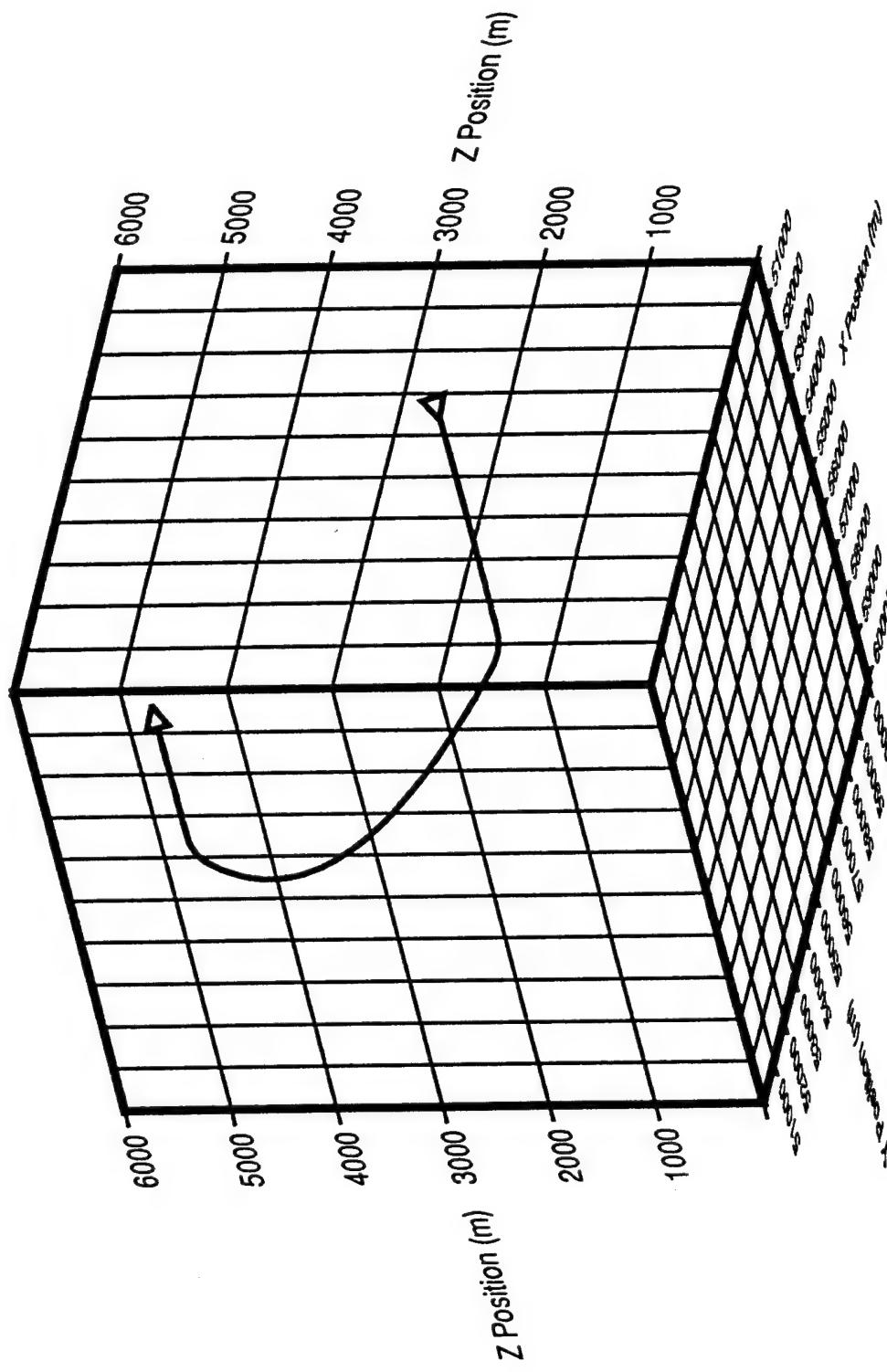
The assumed aircraft track as shown in Figure II-1 consists of an initial level flight leg, a banked and pitched turn of 180° , and a second level flight leg as shown in Figure II-1. The aircraft is initially at 10,000 ft and ascends at a constant pitch of 20° while in the turn. The aircraft roll angle in the turn is -45° . The aircraft yaw rate ($^\circ/\text{sec}$) during the turn is denoted $\dot{\phi}_3$. We note that $\dot{\phi}_3 \bar{v}$ is the magnitude of the acceleration implied by the aircraft yaw rate $\dot{\phi}_3$ and speed \bar{v} . We have taken $\dot{\phi}_3 = 5^\circ/\text{sec}$, so that the 180° turn requires 36 seconds to complete. A 500 mi / hr aircraft speed and a $5^\circ/\text{sec}$ yaw rate in the turn corresponds to approximately a $2g$ aircraft acceleration. The aircraft altitude upon completing the turn is approximately 19,000 ft.

II-1.2 Sensor Characteristics. The characteristics of the low resolution sensor are representative of those of a military air traffic control radar or a shipboard radar and are treated as fixed¹. The characteristics of the high resolution optical sensor are varied parametrically. The principal value shown below for each parameter represents the base case. Alternate variations are shown in parentheses.

¹ Such radars scan in azimuth and do not measure target elevation angle. However, for present purposes we have given the low resolution radar a minimal capability ($\sigma_E = 2^\circ$) to measure elevation angle.

FIGURE II-1

Ground Truth Aircraft Track



Low Resolution Sensor

Measurement errors:

$$\sigma_{Az} = 0.5^\circ$$

$$\sigma_E = 2.0^\circ$$

$$\sigma_R = 200 \text{ meters}$$

Tracking rate:

$$\beta_{LR} = 0.2 \text{ Hz}$$

High Resolution Sensor

Measurement errors:

$$\sigma_{Az} = 0.1^\circ \ (0.05^\circ, 0.2^\circ)$$

$$\sigma_E = 0.1^\circ \ (0.05^\circ, 0.2^\circ)$$

Tracking rate²:

$$\beta_{HR} = 2 \text{ Hz} \ (1 \text{ Hz}, 4 \text{ Hz}).$$

II-1.3 Image Templates. The image template library for the X29 was provided by WUSL. The template resolution is 64×64 pixels with the reflected visible light intensity level represented on a gray scale indexed from 0 to 255. Roll and pitch angles are discretized uniformly in 10° increments. The yaw angle increment varies with pitch angle. The number of yaw values is maximum at 32 (a step size 11.25°) when the pitch angle is 0° . The yaw discretization becomes increasingly coarse as pitch increases until there is only a single yaw value of 0° at a pitch of 90° .

Observed images are simulated by adding (truncated) Gaussian white noise to the library templates. The scale of the image noise is measured in terms of a number of gray scale intensity levels. The assumed base case pixel noise level is $\sigma_N = 12$ gray scale units (gsu).

² The actual frame rate of the optical sensor may be 20 Hz - 60 Hz. However, we assume a much lower effective tracking rate to reflect any multi-frame integration inherent in the signal processing and also to justify our assumption that angle measurement errors are statistically independent and that the pixel noise in the optical images is white.

The base case parameter values and variations used in the image simulation and processing are summarized as follows:

pixel resolution = 64×64 (32×32 , 16×16)

gray scale levels = 256 (128, 64)

pixel noise = 12 gsu (24 gsu, 48 gsu).

As shown in Figure II-1, the two sensors are collocated at a point which is 50 miles (measured horizontally) from the aircraft at the midpoint of its turn. Low resolution tracking begins at $t=0$. The high resolution sensor begins reporting at $t=20$ secs, and from that point on tracking is based on fusing the low resolution and high resolution data. The aircraft 180° turn is initiated at $t=30$ secs and completed at $t=66$ secs, at which point the tracking problem is terminated.

We analyzed three tracking variations based on which sensor data are fused into the track solution:

(LR): Low resolution position measurements only

(HRA): Low resolution position measurements + high resolution angle measurements

(HRI): Low resolution position measurements + high resolution angle measurements + high resolution pixel images.

The track estimation in the LR and HRA processing variations is based on the Kalman filter described in Chapter I. In the HRI variation, the image data is added and the jump-diffusion model approach (also described in Chapter I) is used.

II-2 Numerical Results

Figures II-2 through II-11 below represent the principal numerical results of our Phase I study. Figures II-2 through II-6 apply to the base case and show in succession: error in position, velocity, roll angle, pitch angle, and yaw angle. In the case of position error and velocity error, the LR and HRA processing results are shown for comparison.

The position tracking results in Figure II-2 show that during the period from $t = 20$ secs to $t = 30$ secs, while the aircraft is still in level flight, the fusion of the high resolution angle data with the low resolution position data has a very significant effect. The tracking error rises to as high as 3,000 meters if only the radar data are processed. If the optical sensor angle measurements are fused into the tracking solution, the tracking error is kept under 500 meters.

Once the aircraft maneuver begins at $t = 30$ secs, even the combination of radar data and optical angle data is not enough to keep the tracking error from increasing from about 200 meters at the beginning of the maneuver to as much as 1,400 meters during the maneuver. One observes that in this HRA processing case the position error exhibits a very distinct sawtooth behavior with a 5 sec period. The position error drops sharply each time the radar reports a target range measurement and rises systematically between successive radar measurements. This behavior reflects the fact that the optical sensor provides no direct target range information and therefore the component of position error along the line of sight tends to grow between radar measurements.

Once the image data are fused into the tracking solutions, the situation changes rather markedly. Now the target orientation information deduced from the target images can be used to deduce the direction of the target forward motion. As a result, knowledge of the target position along the line of site does not degrade between radar observations. One observes in Figure II-2 that the sawtooth behavior prominent in the HRA case is now much less in evidence. The net result is that position error under HRI processing is consistently less than under HRA processing and remains in the range 200 meters to 600 meters, about the same as during the period the target was in level flight.

Figure II-3 shows the base case velocity (vector) error comparison. Again the benefit of using imagery data in support of tracking is in evidence. During the interval from $t = 20$ secs. to $t = 30$ secs., while the target is still in level flight, velocity error is consistently less than 30 m/sec under HRA processing. Once the target starts its maneuver, there is a more or less systematic increase in velocity error until about 40 secs. both under HRA and HRI processing. This early stage of the maneuver presents a particularly difficult tracking problem since two and sometimes all three of the target orientation angles are changing simultaneously. Once the target roll and pitch steady to their new values and only the target yaw angle is changing, velocity error trends generally downward under HRI processing. Under HRA processing, velocity error trends generally upward until it peaks at about 200 m/sec at $t = 50$ secs. (the approximate midpoint time of the target maneuver) before starting a slow decline.

FIGURE II-2

Base Case: Position Error

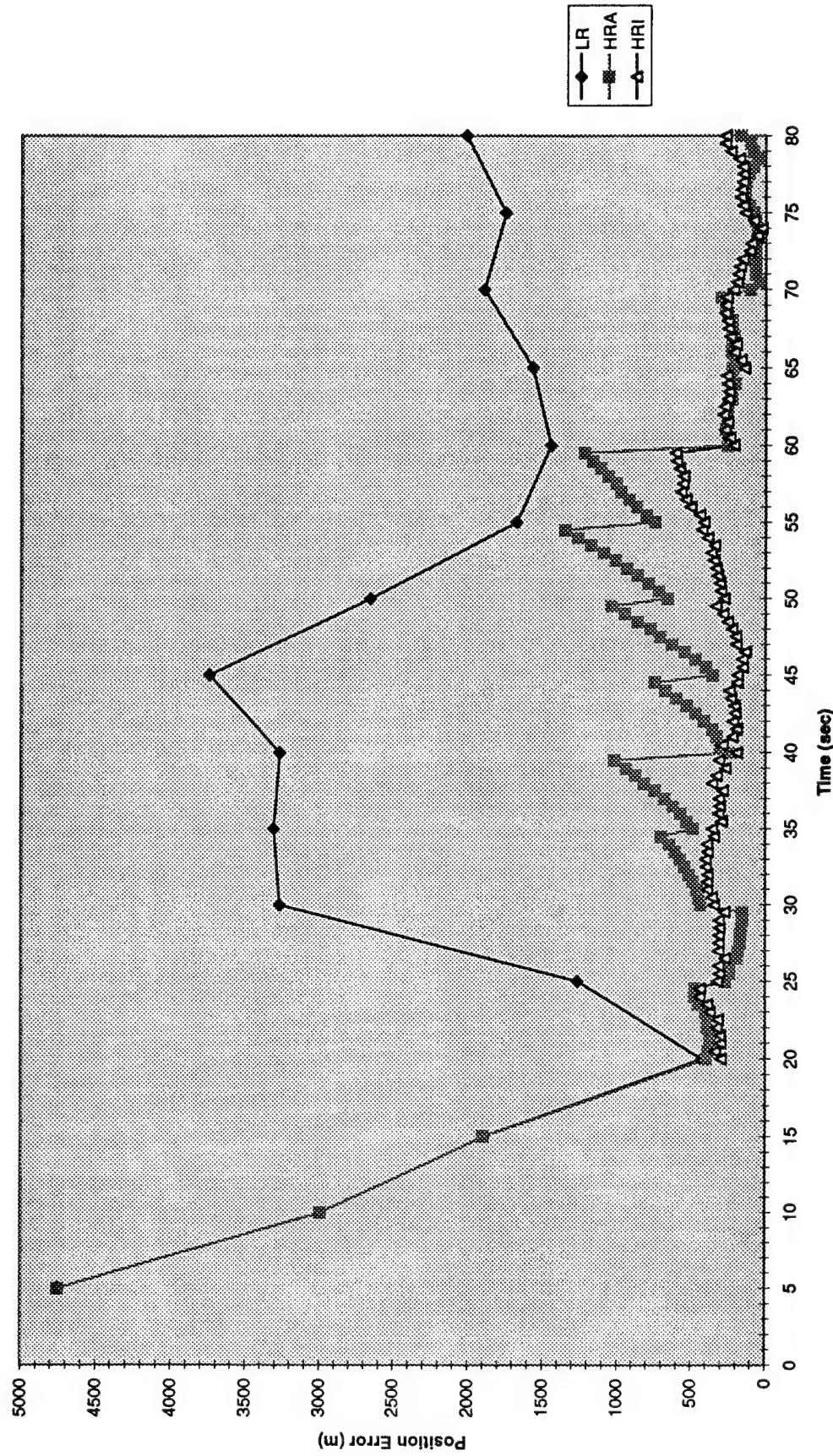
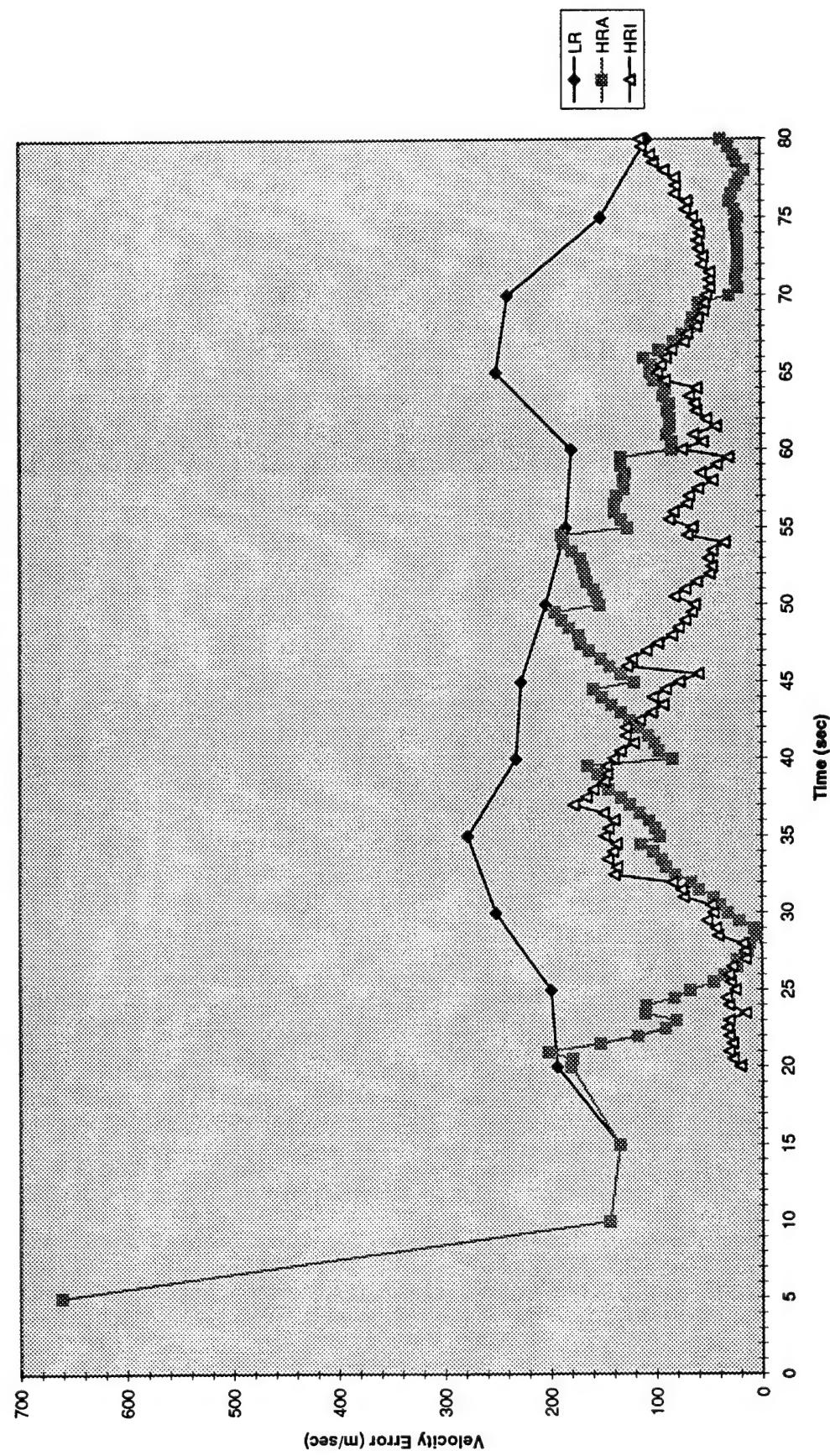


FIGURE II-3

Base Case: Velocity Error



Figures II-4 through II-6 show the behavior in the base case of the errors in estimates of the target orientation angles under HRI processing. One observes that yaw angle error tends to be larger in magnitude and more variable than either roll angle error or pitch angle error. One contributing factor is that once the target maneuver starts, the target yaw angle varies continuously at the rate of $5^\circ/\text{sec}$ throughout the 36 sec. period of the maneuver. Roll and pitch, on the other hand, stabilize at a constant value within 9 secs. There is also the effect that the discretization of yaw angle is variable with pitch angle and generally coarser than the discretization of roll and pitch.

We summarize our base case numerical results as follows:

- **Case I:**
 - (i) target not maneuvering
 - (ii) fusion of low resolution (radar) and high resolution (optical) point tracking data (HRA processing)
 - (iii) Kalman filter state estimation methods.

Then tracking errors vary from 200 meters to 500 meters. This compares with tracking errors of as much as 3,000 meters under low resolution (LR) point tracking.

- **Case II:**
 - (i) target maneuvering ($2g$ acceleration for 36 secs.)
 - (ii) fusion of low resolution point tracking data, high resolution point tracking data, and imaging data (HRI processing)
 - (iii) jump-diffusion method of state estimation.

Then tracking errors vary from 200 meters to 600 meters. This compares with tracking errors of as much as 1,400 meters under high resolution (HR) point tracking.

FIGURE II-4

Base Case: Roll Error

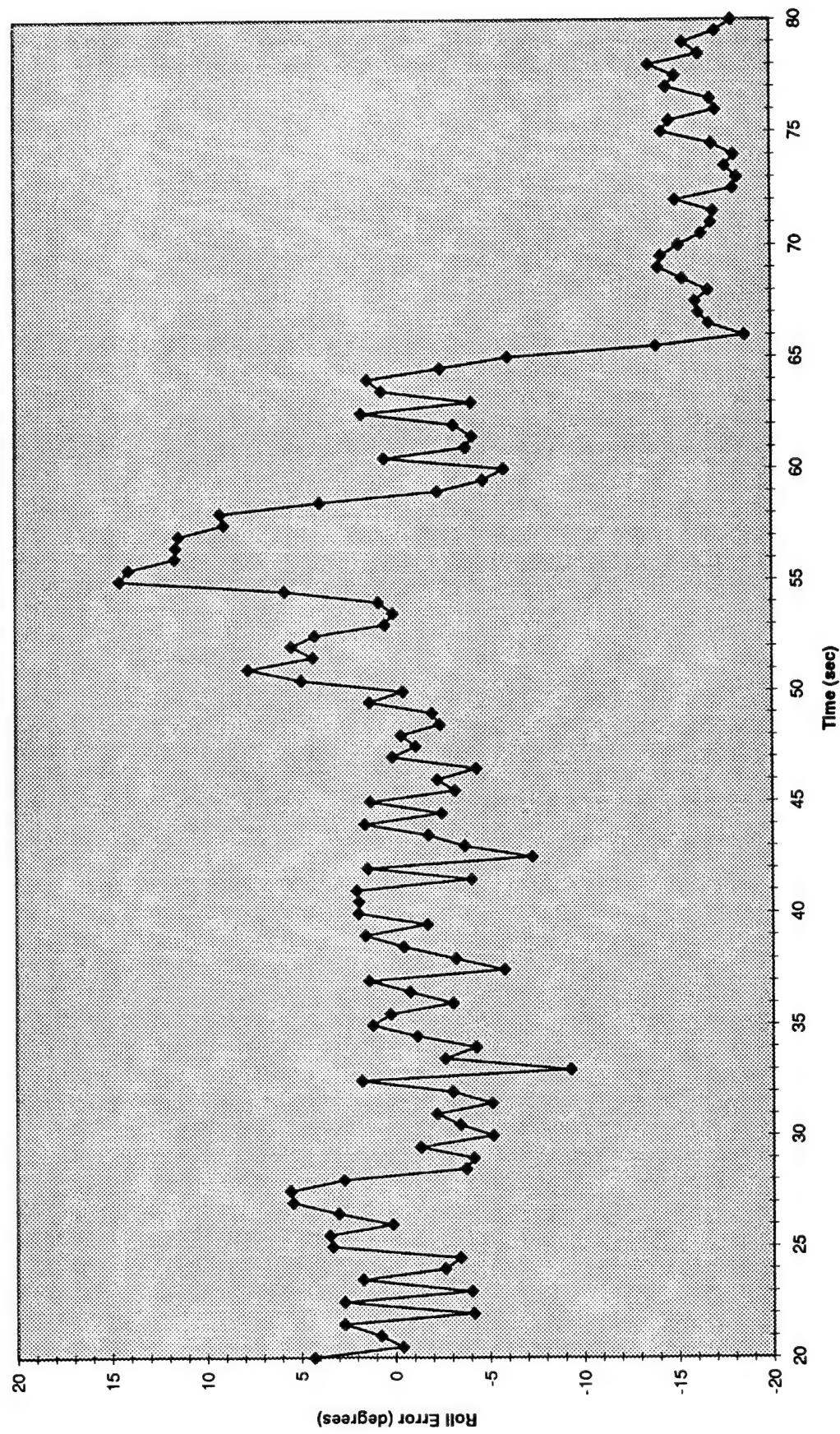


FIGURE II-5
Base Case: Pitch Error

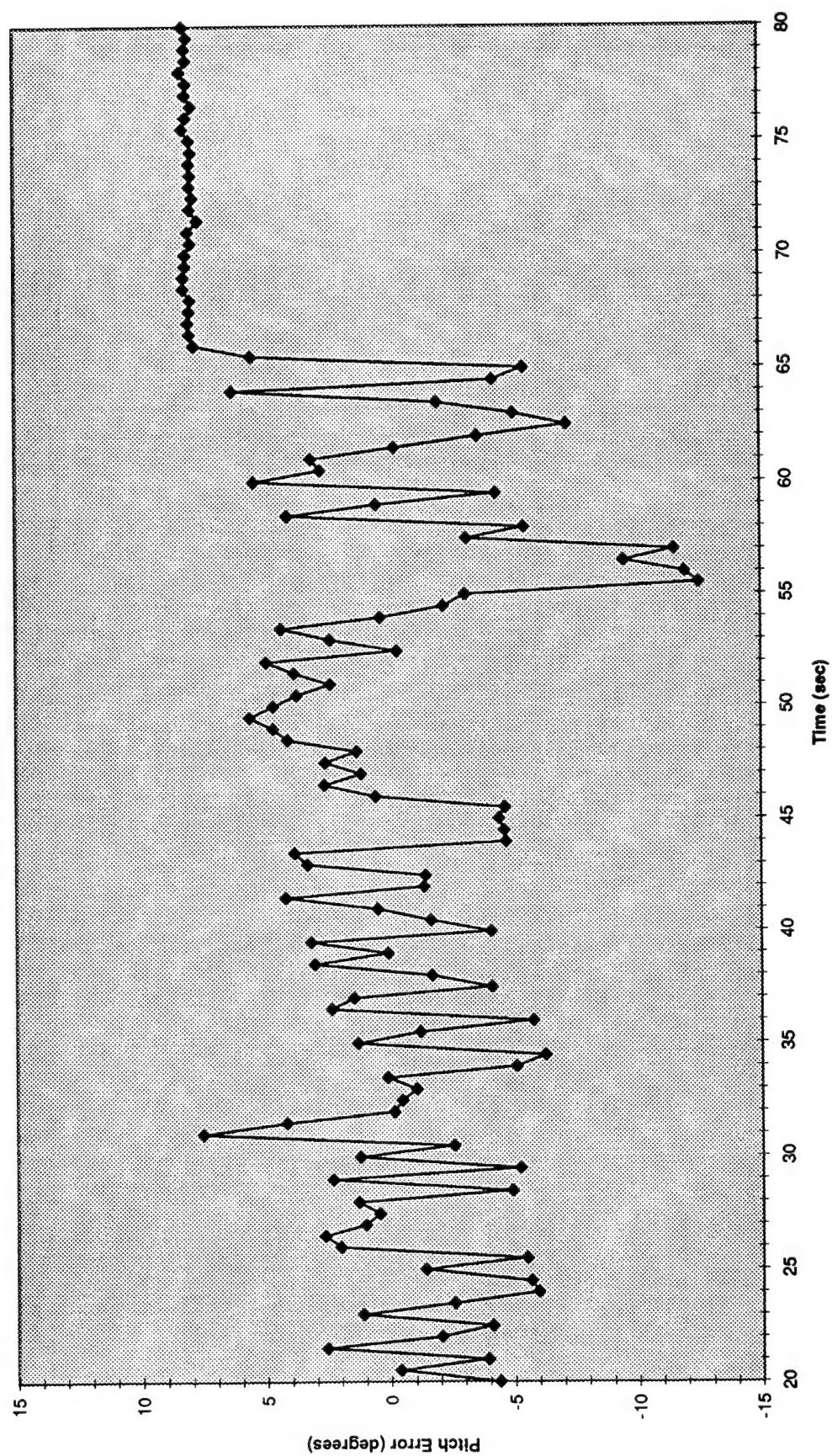
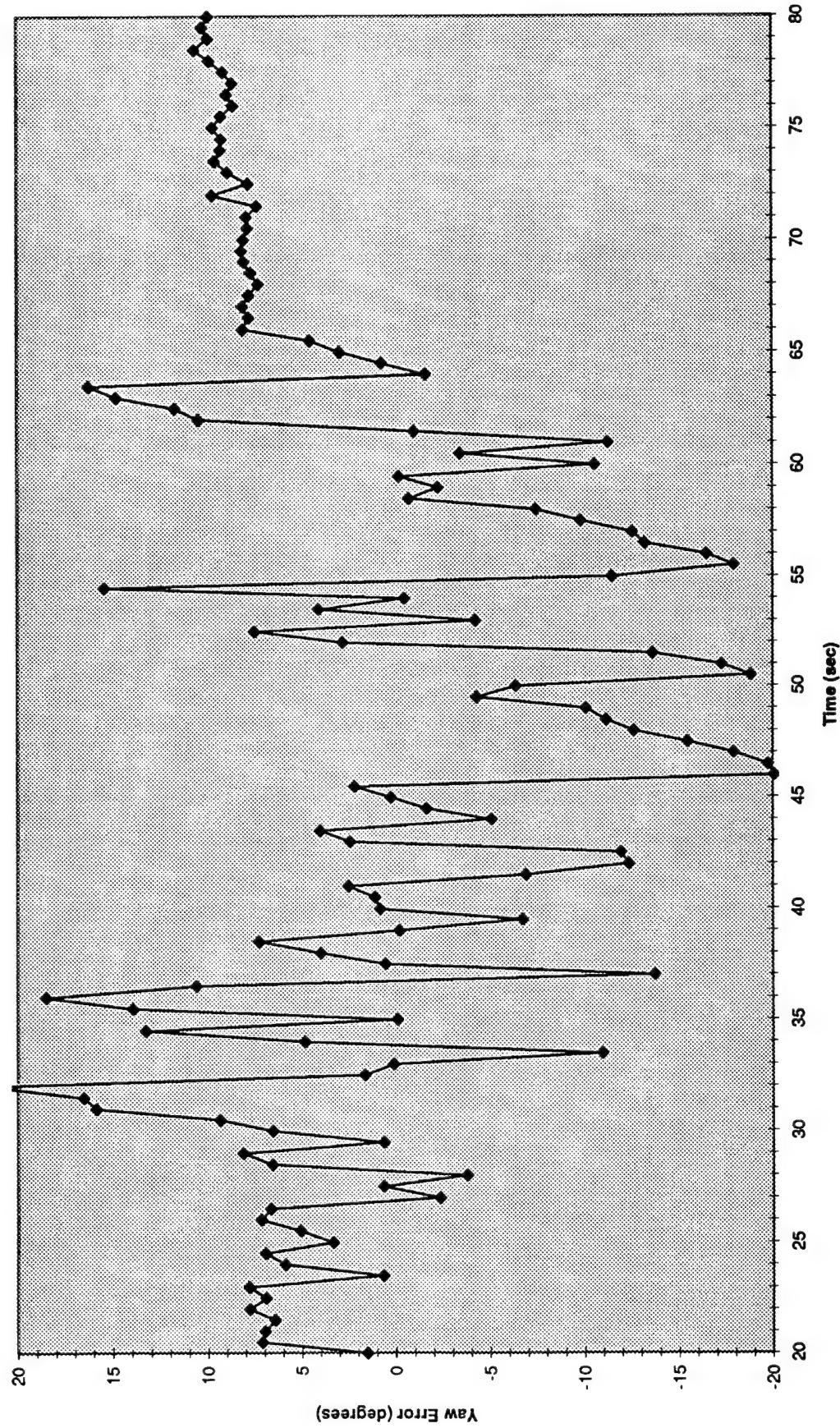


FIGURE II-6

Base Case: Yaw Error



Figures II-7 through II-11 show the effect on tracking error of varying five of the key problem parameters relative to their base case values:

- optical sensor direction measurement error
- optical sensor measurement rate
- number of pixels per target image
- number of pixel intensity levels
- pixel noise level.

In each case, tracking error is plotted out to the midpoint of the aircraft maneuver ($t = 48$ secs). Each figure shows three plots: the base case and two variations of the indicated parameter.

Figures II-7 and II-8 show that there is a marked sensitivity of tracking error to the data rate and measurement error of the optical sensor. One observes, in particular, that a factor of two variation either in data rate or direction measurement error can induce a factor of two variation in tracking error.

Figure II-9 shows the effect of varying the number of pixels used to represent an aircraft image. We note that as the number of pixels is decreased, an adjustment is made in the pixel noise level to account for the aggregation of smaller pixels into larger pixels. For example, each pixel in a 32×32 image is the composite of 4 pixels in the corresponding 64×64 representation, such that the intensity level assigned to the composite pixel is the average of the intensities of the 4 component pixels. This averaging has the statistical effect of reducing the pixel noise level by a factor of two.

One observes from Figure II-9 that the relationship between the level of pixel aggregation and tracking error appears to be a complex one. Prior to about 36 secs, tracking error is minimal when 32×32 pixel images (as opposed to 16×16 pixel images or 64×64 pixel images) are used. Beyond about 42 secs, tracking error is minimal when 64×64 images are used. Between 36 secs and 42 secs, the three pixel aggregation levels result in about the same tracking error.

A possible explanation for this observed behavior is that the optimal degree of pixel aggregation depends on the target viewing geometry. More pixel aggregation implies less image detail but reduced noise in each pixel. Less aggregation implies more image detail but increased noise in each pixel. The best operating point is likely to vary with the amount of structural detail of the target that is needed to resolve its orientation. This in turn varies with the target viewing geometry.

FIGURE III-7

Measurement Error Variations

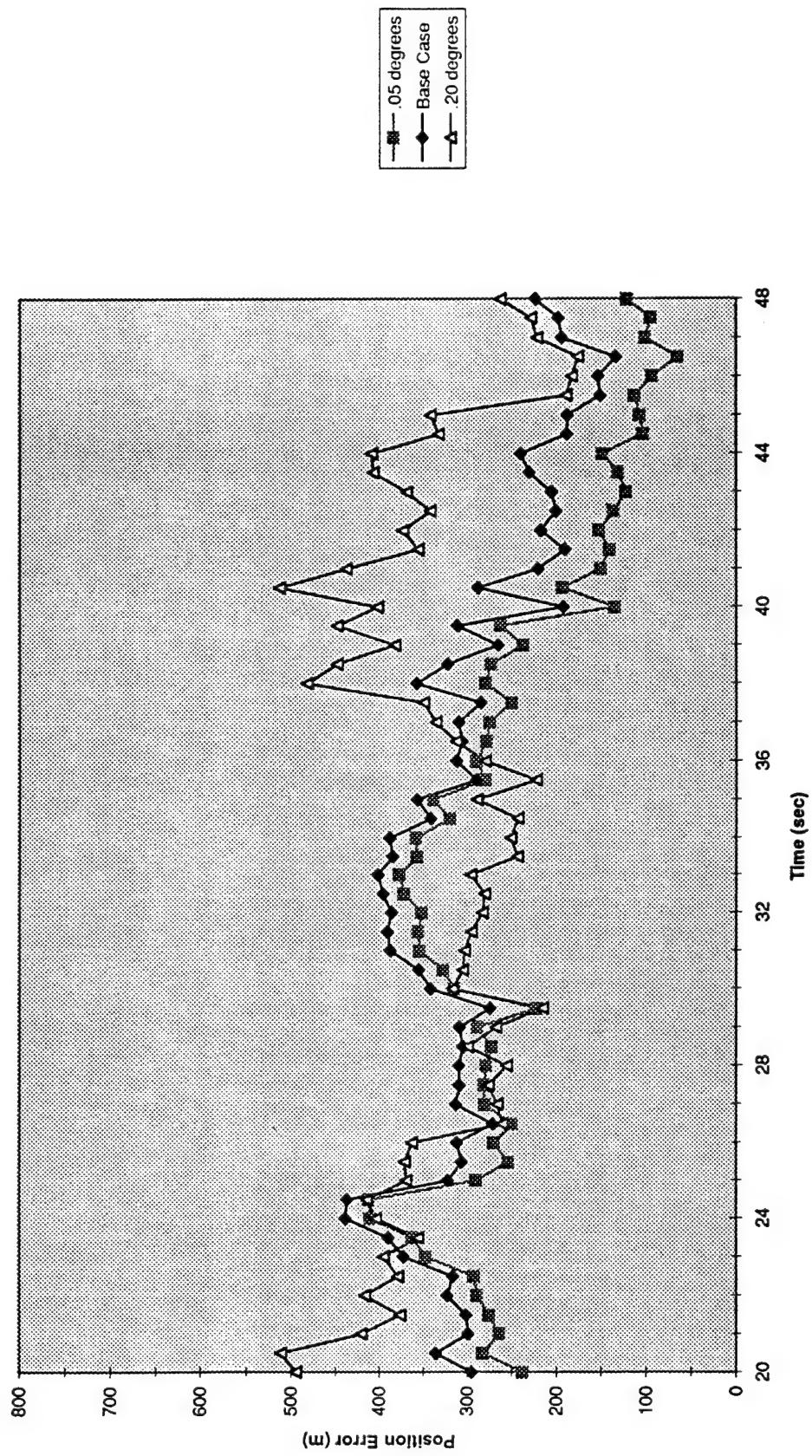


FIGURE II-8

Data Rate Variations

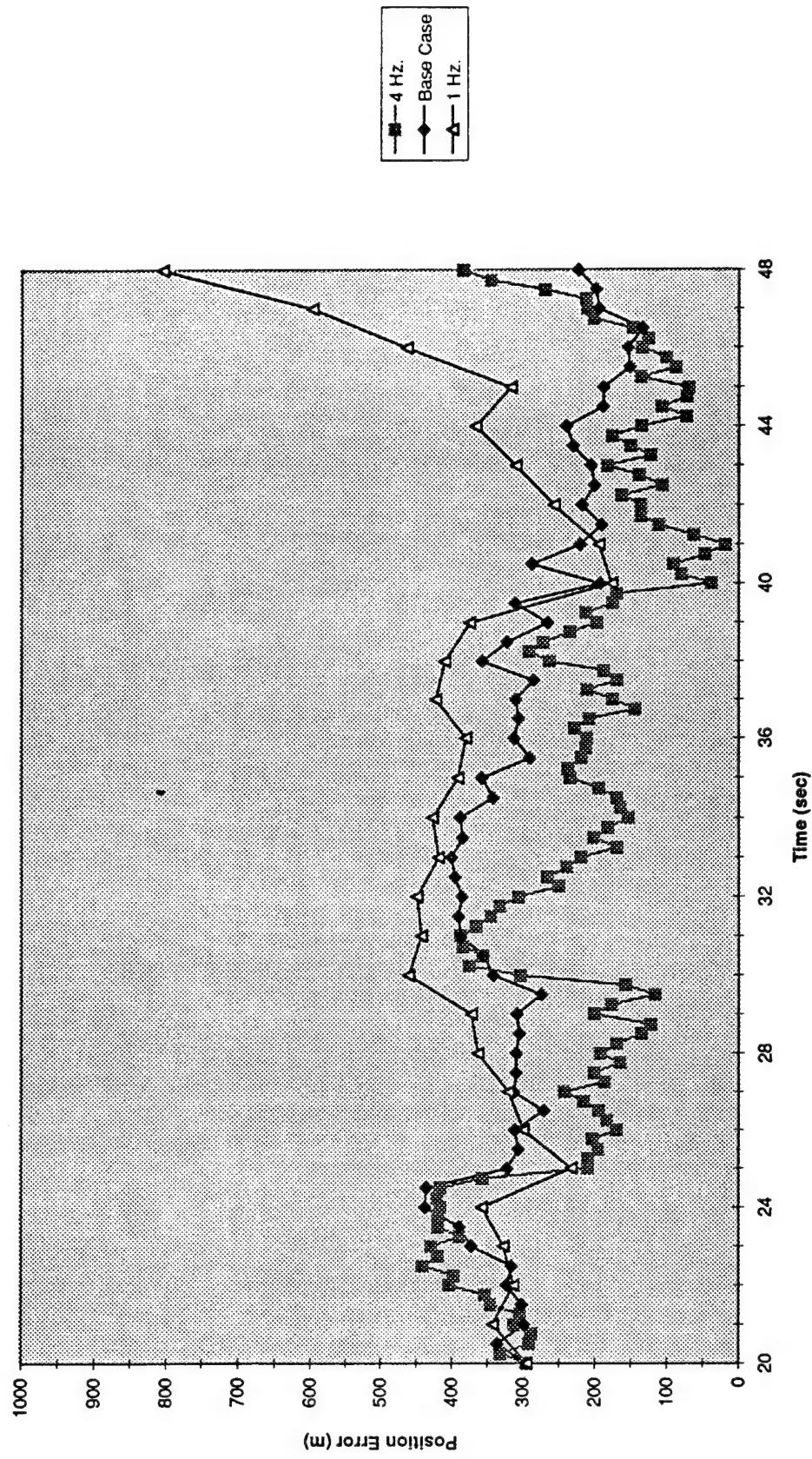


FIGURE III-9

Variations in Number of Pixels

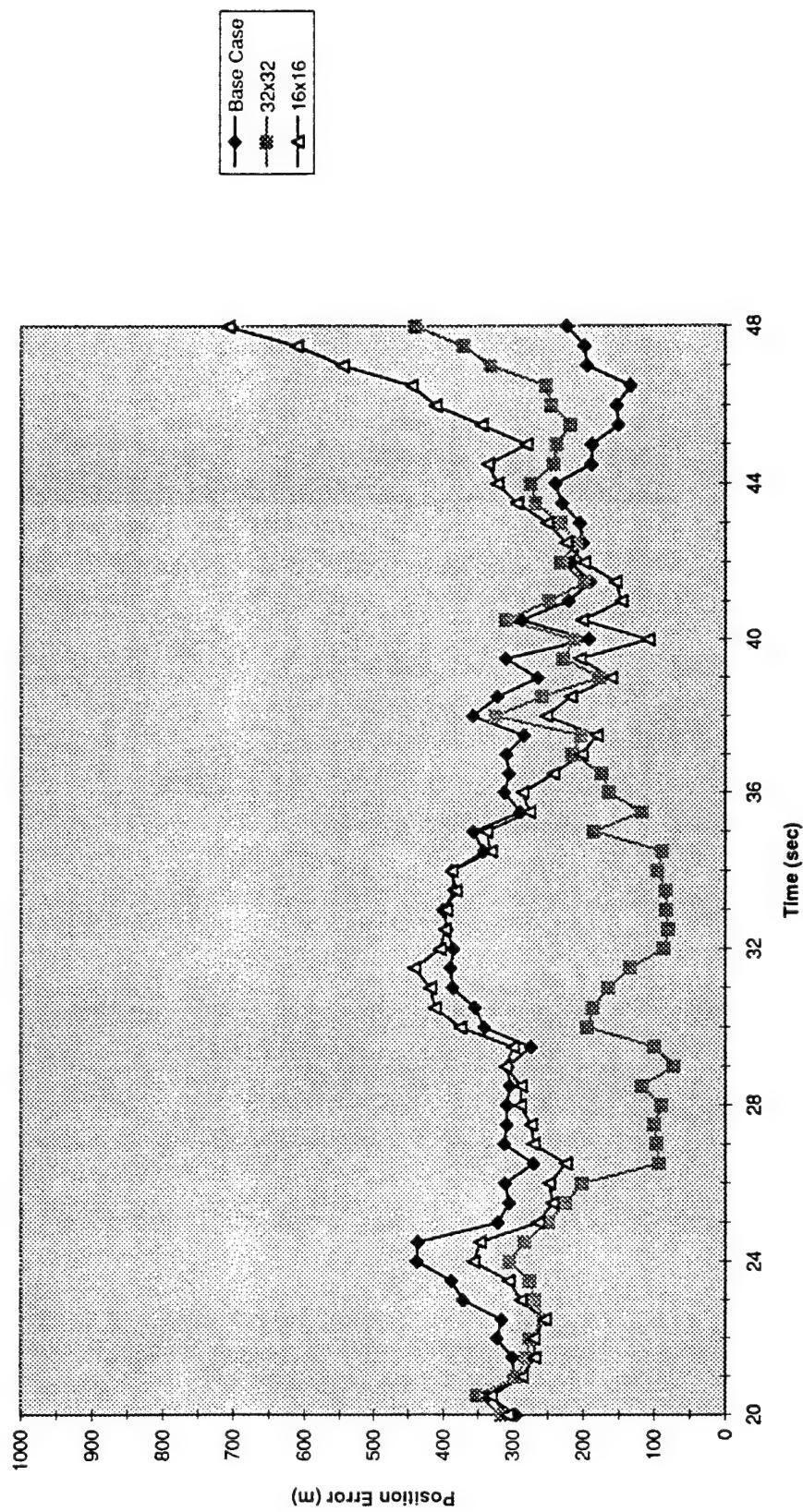


Figure II-10 shows the result of decreasing the number of distinct gray scale levels. As in the pixel aggregation case, it is appropriate to adjust the pixel noise level as the number of gray scale levels is varied. In the present case, the pixel error was adjusted to maintain a constant ratio between the noise level measured in gray scale units and the number of gray scale levels.

Figure II-10 shows that tracking error consistently increases as the number of gray scale intensity levels decreases. Furthermore, the effect is typically a rather pronounced one.

Figure II-11 shows the effect of varying the level of pixel noise. One observes that there is something of a threshold effect at work, in that the increase in noise from 12 gsu to 24 gsu has only a minimal effect on tracking error. However, the further increase in pixel noise from 24 gsu to 48 gsu has a marked impact in terms of increased tracking error.

We summarize our observations based on our parametric study as follows:

- Tracking error shows a strong dependence on the measurement accuracy and data rate of the optical sensor
- A decrease in the number of gray scale intensity levels and an increase in the level of pixel noise each cause the tracking error to increase. The magnitude of the effect is small, however, in the case of increasing the pixel noise from 12 gsu to 24 gsu
- The effect of varying the number of pixels per image appears to be a complex one, suggesting a possible geometry-dependent tradeoff between image detail and effective pixel noise level.

We have made no systematic effort in Phase I to benchmark the fusion algorithm execution time. Optimization of the algorithm implementation in order to meet operational execution time constraints will be a Phase II technical objective. We do observe, however, that the execution time per measurement update varies approximately linearly with

$$N = (\text{number of Monte Carlo sample tracks}) \\ \times (\text{number of diffusion cycles per measurement update}).$$

FIGURE II-10

Variations in Number of Intensity Levels

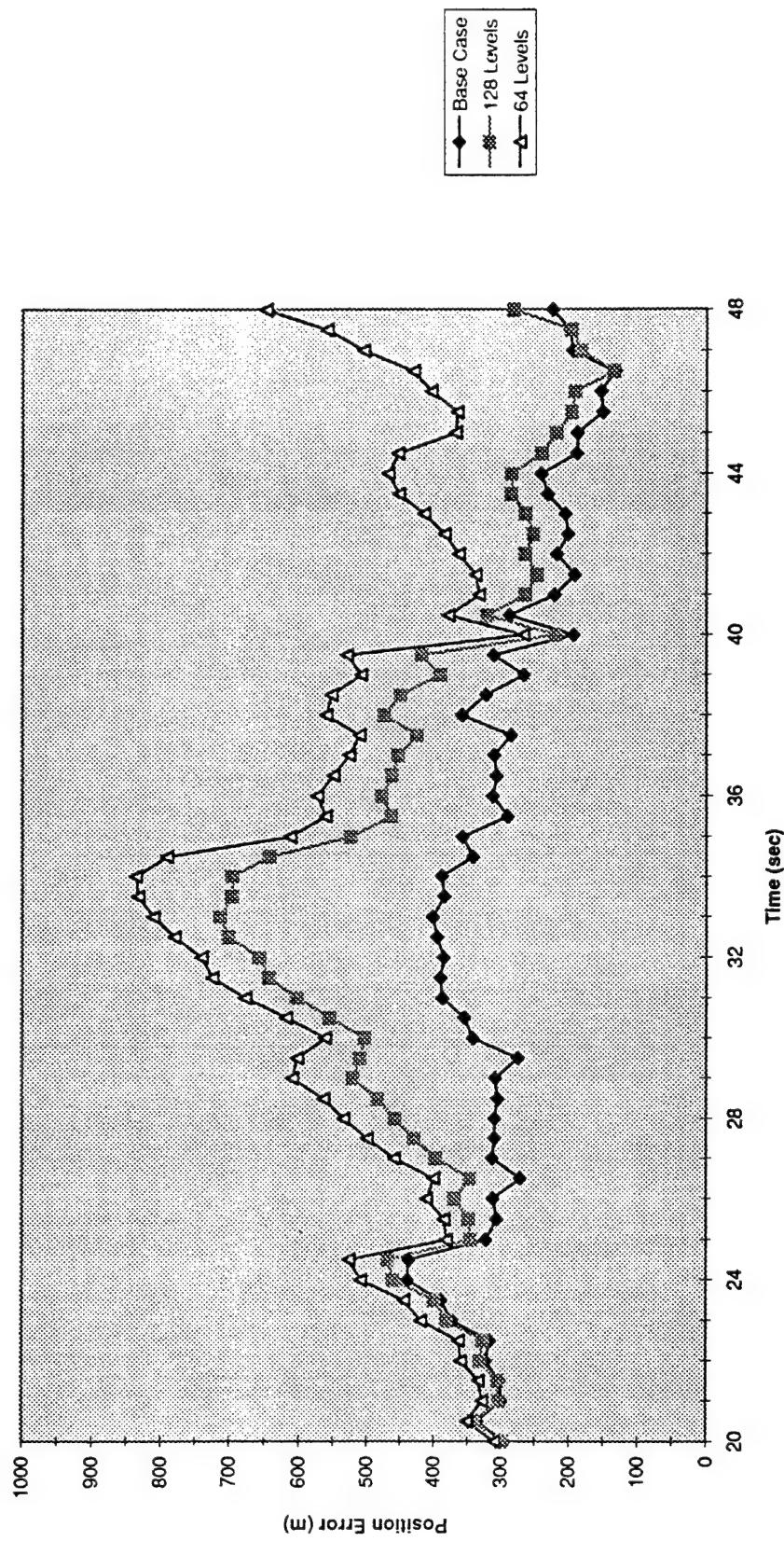
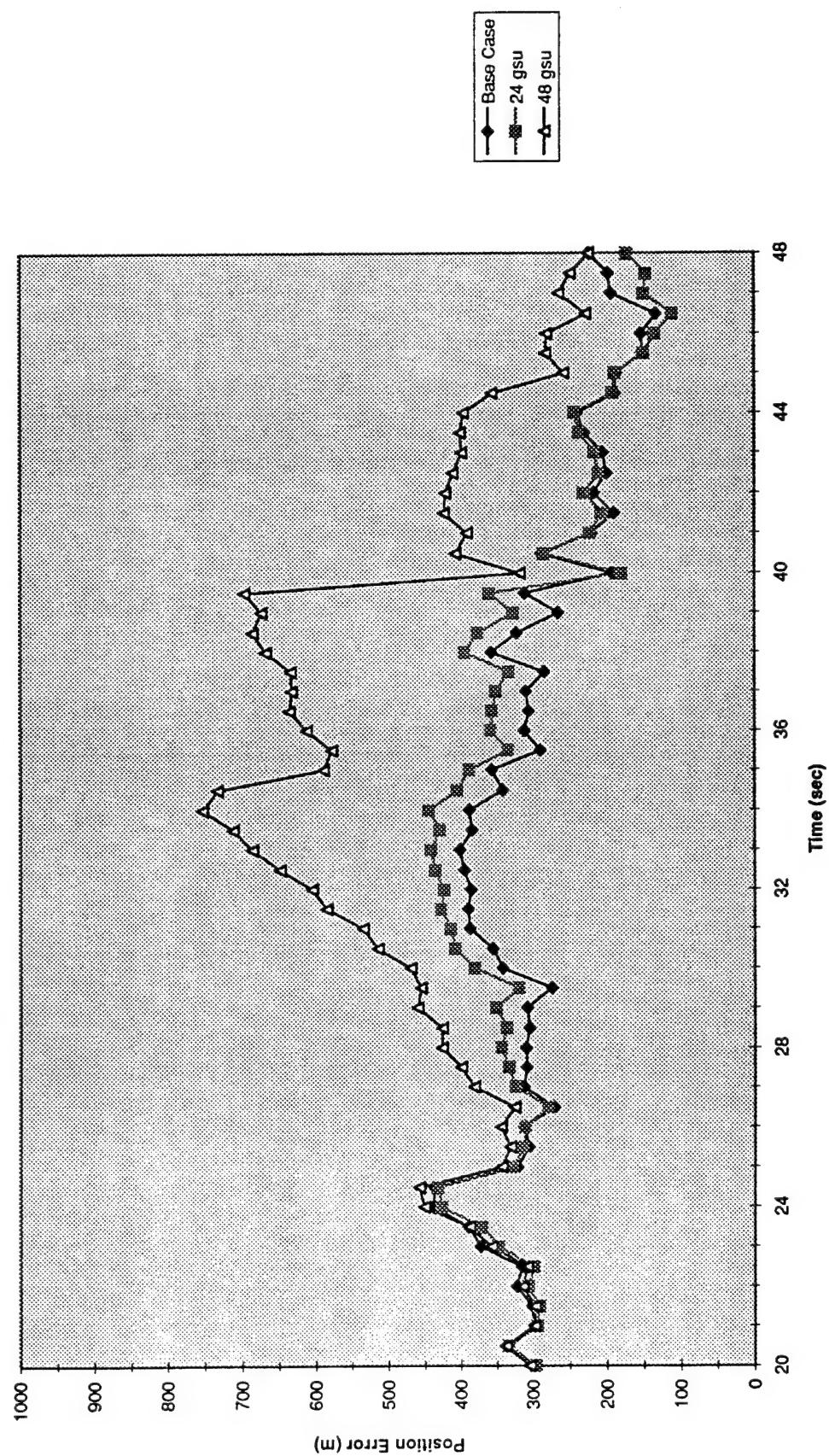


FIGURE III-11

Pixel Noise Variations



In our Phase I testing, we used $N = 100 \times 2,000 = 200,000$, a value deliberately chosen generously high from the standpoint of the statistical convergence of track estimates. In Phase II we will develop improved methods for choosing the step size per cycle in an effort to reduce significantly the number of diffusion cycles required. We will then investigate in detail the trade-offs between the statistical accuracy of the algorithm estimates and execution time. The eventual operational goal in Phase III will be to achieve near real-time processing on the designated computing platform.

II-3 Study Conclusions

Our principal Phase I study conclusions are:

- We successfully demonstrated that the computations required to carry out the data fusion algorithm operations using our estimation methods, previously executed only within the context of a massively parallel computer architecture, can be performed on a serial processing workstation (the equivalent of a Sun Microsystems SPARC 20).
- Our parametric study shows that there are significant benefits to be gained in terms of increased tracking accuracy through the fusion of multisensor point tracking and target image data. We have demonstrated this specifically for the combination of a low resolution radar sensor and a high resolution optical imaging sensor. However, the methods we have developed to fuse multisensor data are quite general and can be extended to other sensing technologies.

References

- [1] "Jump-Diffusion Processes for Abduction and Recognition of Biological Shapes," U. Grenander and M. I Miller, *Monograph of the Electronic Signals and Systems Research Laboratory*, 1991.
- [2] "Automated Target Tracking and Recognition Using Jump Diffusion Processes," A. Srivastava, M.S. Thesis, Washington University, St. Louis, Missouri, December 1993.
- [3] "Automated Target Recognition in a Distributed Computing Environment," R. S. Teichman, M.S. Thesis, Sever Institute of Washington University, St. Louis, Missouri, December 1994.
- [4] "Pattern Theoretic Bayesian Inference for Multisensor Fusion," M. I. Miller, A. Lanterman, and A. Srivastava.
- [5] "A New Approach for Tracking of Multiple, Airborne Targets," M. I. Miller and A. Srivastava.
- [6] "Equation of State Calculations by Fast Computing Machines," N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, *Journal of Physical Chemistry*, 21, 1087, 1953.
- [7] "Control System Design: An Introduction to State-Space Methods," B. Friedland, McGraw-Hill Book Company, 1986.

APPENDIX

Aircraft Dynamics

In this Appendix we record certain formulas related to the dynamics of aircraft motion. These formulas are useful in constructing ground truth aircraft tracks and in determining the accelerations implied by hypothesized aircraft maneuvers.

Consider the aircraft body frame which has the coordinate axes fixed along the principal axes of the airframe and define

$$\mathbf{v}_B = (v_1^B, v_2^B, v_3^B)$$

= aircraft velocity vector projected onto body frame

$$\mathbf{q}_B = (q_1^B, q_2^B, q_3^B)$$

= aircraft angular velocity vector projected onto body frame

$$\mathbf{f}_B = (f_1^B, f_2^B, f_3^B)$$

= vector of forces acting on aircraft in the body frame.

Rigid body dynamics requires that the following system of differential equations must hold at any time s (see Section 2.6 of [7]):

$$\begin{aligned}\dot{v}_1^B(s) - q_3^B(s)v_2^B(s) + q_2^B(s)v_3^B(s) &= f_1^B(s) \\ \dot{v}_2^B(s) + q_3^B(s)v_1^B(s) - q_1^B(s)v_3^B(s) &= f_2^B(s) \\ \dot{v}_3^B(s) - q_2^B(s)v_1^B(s) + q_1^B(s)v_2^B(s) &= f_3^B(s).\end{aligned}\tag{A-1}$$

We note that (A-1) can be expressed in vector form as $\dot{f}_B = \dot{v}_B + q_B \times v_B$, where the term $q_B \times v_B$ represents the Coriolis acceleration induced by the rotation of the body frame relative to a reference inertial frame.

The rotational velocity vector (q_1^B, q_2^B, q_3^B) in the body frame is equal to the vector $(\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3)$, where

$$\begin{aligned}\dot{\phi}_1 &= \text{roll rate} \\ \dot{\phi}_2 &= \text{pitch rate} \\ \dot{\phi}_3 &= \text{yaw rate.}\end{aligned}$$

The angular velocity (q_1, q_2, q_3) in the inertial frame is related to the Euler angles by

$$\begin{aligned}q_1(s) &= \dot{\phi}_1(s) - \dot{\phi}_3(s)\sin(\phi_2(s)) \\ q_2(s) &= \dot{\phi}_2(s)\cos(\phi_1(s)) + \dot{\phi}_3(s)\cos(\phi_2(s))\sin(\phi_1(s)) \\ q_3(s) &= -\dot{\phi}_2(s)\sin(\phi_1(s)) + \dot{\phi}_3(s)\cos(\phi_2(s))\cos(\phi_1(s)).\end{aligned}\quad (\text{A-2})$$

The inertial position vector is given by

$$p(s) = \int_{t_0}^s \Psi^T(\tau) v_B(\tau) d\tau + p(t_0), \quad (\text{A-3})$$

where the transformation

$$\begin{aligned}\Psi(\tau) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_1(\tau)) & \sin(\phi_1(\tau)) \\ 0 & -\sin(\phi_1(\tau)) & \cos(\phi_1(\tau)) \end{bmatrix} \times \begin{bmatrix} \cos(\phi_2(\tau)) & 0 & -\sin(\phi_2(\tau)) \\ 0 & 1 & 0 \\ \sin(\phi_2(\tau)) & 0 & \cos(\phi_2(\tau)) \end{bmatrix} \\ &\quad \times \begin{bmatrix} \cos(\phi_3(\tau)) & \sin(\phi_3(\tau)) & 0 \\ -\sin(\phi_3(\tau)) & \cos(\phi_3(\tau)) & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}\quad (\text{A-4})$$

rotates the inertial frame axes into the body frame axes.

If one assumes that the aircraft velocity vector in the body frame points in the positive x-axis direction, then $v_B = (\bar{v}, 0, 0)$, where \bar{v} is the aircraft speed. If \bar{v} is assumed constant, then the relationship $f_B = \dot{v}_B + q_B \times v_B$ can be used to relate a specified combination of roll rate, pitch rate, and yaw rate to the implied accelerations. It then becomes a straightforward matter to design aircraft maneuvers involving accelerations of a specified number of g 's, where g is the gravitational acceleration. If the assumed Euler angle rates are assumed constant, one can directly perform the integration in Equation (A-3) to determine the aircraft position as a function of time.